



TESIS – KI142502

**PENGEMBANGAN METODE *REDUCED DIFFERENCE*
EXPANSION PADA *DATA HIDING* DENGAN TEKNIK
INTERPOLASI CITRA**

**LIDYA AMALIA RAHMANIA
NRP: 5114201038**

**DOSEN PEMBIMBING
Tohari Ahmad, S.Kom, MIT., Ph.D**

**PROGRAM MAGISTER
BIDANG KOMPUTASI BERBASIS JARINGAN
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

[Halaman ini sengaja dikosongkan]

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

LIDYA AMALIA RAHMANIA

NRP 5114201038

Dengan judul:

Pengembangan Metode Reduced Difference Expansion pada Data Hiding dengan
Teknik Interpolasi Citra

Tanggal Ujian : 13 Januari 2017

Periode Wisuda : 2017 Gasal

Disetujui oleh:

Tohari Ahmad, S.Kom, MIT, Ph.D
NIP. 19750525 2003 12 1 002

(Pembimbing 1)

Waskitho Wibisono, S.Kom, M.Eng, Ph.D.
NIP. 19741022 200003 1 001

(Penguji 1)

Royyana Muslim I., S.Kom., M.Kom., PhD.
NIP. 19770824 200604 1 001

(Penguji 2)

Henning Titi C., S.Kom., M.Kom.
NIP. 19840708 201012 2 004

(Penguji 3)

Direktur Program Pasca Sarjana,



Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 1960120 2198701 1 001

[Halaman ini sengaja dikosongkan]

PENGEMBANGAN METODE *REDUCED DIFFERENCE EXPANSION* PADA DATA *HIDING* DENGAN TEKNIK INTERPOLASI CITRA

Nama Mahasiswa : Lidya Amalia Rahmania
NRP : 5114201038
Pembimbing : Dr. Tohari Ahmad, S.Kom, MIT.

ABSTRAK

Dewasa ini penggunaan internet sebagai media untuk bertukar informasi menjadi hal yang lazim dilakukan. Sebagai salah satu jalur komunikasi utama yang kerap digunakan, maka perlindungan terhadap data yang ditransmisikan melalui jaringan internet harus diaktifkan. Salah satu usaha untuk melindungi data yang sensitif adalah dengan teknik steganografi.

Saat ini metode *data hiding* yang banyak dikembangkan adalah *Difference Expansion* (DE). Kekurangan dari metode ini adalah diperlukan adanya sebuah *location map* untuk menentukan posisi bit data rahasia yang ada dalam sebuah piksel tertentu. Hal ini dikarenakan tidak semua piksel dapat menyimpan bit data dari pesan rahasia karena nilainya yang terlalu tinggi atau terlalu rendah.

Metode lain yang dikembangkan dari metode DE adalah *Prediction Error Expansion* (PEE). PEE dikembangkan karena tuntutan penambahan kapasitas pesan yang dapat disimpan pada sebuah citra stego. PEE tidak membutuhkan *location map* karena proses penyimpanan pesan dilakukan secara sekuensial.

Penelitian ini mengusulkan metode data hiding yang digabungkan dengan teknik interpolasi pada citra digital untuk menambahkan kapasitas data rahasia yang disimpan. Dengan menggunakan sebuah citra digital yang dibagi menjadi blok piksel tidak tumpang tindih berukuran 3×3 piksel, kemudian dilakukan interpolasi untuk mendapatkan prediksi nilai baru piksel. Pesan rahasia dimasukkan ke dalam citra yang baru menggunakan dua kali proses penyisipan. Proses yang pertama dilakukan penyisipan pada piksel yang nilai barunya berasal dari proses interpolasi, sedangkan proses penyisipan yang kedua dilakukan pada piksel referensi yang tidak diinterpolasi. Diharapkan penelitian ini menghasilkan sebuah metode penyembunyian data yang dapat meningkatkan kapasitas data dalam sebuah citra namun dengan resiko sedikit penurunan kualitas yang mungkin terjadi.

Kata Kunci: *data hiding, reduced difference expansion, prediction error expansion, location map, interpolasi.*

[Halaman ini sengaja dikosongkan]

ENHANCED REDUCED DIFFERENCE EXPANSION DATA HIDING USING IMAGE INTERPOLATION

Name : Lidya Amalia Rahmania
NRP : 5114201038
Supervisor : Dr. Tohari Ahmad, S.Kom, MIT.

ABSTRACT

Nowadays Internet is used as a media for information transaction on a daily basis. As one of main communication line, thus the protection for the transmitted data via Internet must have been done. One way to protect the sensitive data is to perform steganography technique.

The most developed data hiding method is Difference Expansion (DE). The disadvantage of this method is the requirement to make a location map to decide which pixels are used to hide the secret bits. This happened because not all of the pixels could be used to hide the secret bits because the value of the pixels are too high or too low.

Another method that developed other than DE is Prediction Error Expansion (PEE). PEE developed to add the message capacity that can be hidden in a stego image. This method hide the secret data bits based on the expectation values of others neighborhood pixels. Then the secret bits embedded to the new pixel values after added by prediction values. PEE does not need location map because the sequential embedding process. The disadvantage of this method is the decreased quality of stego image.

This research proposed a data hiding method that integrated with interpolation technique in digital image to add the embedded secret data capacity. Using a digital image that interpolated based on non-overlapped 3×3 block pixels, the secret bit embedded in the cover image using two time embedding processes. The first embedding process will be done to the result of the interpolated pixels, and the second embedding process will be done to the reference pixels using the uninterpolated method.. This research expected to be a new data hiding method that can improve the data bit capacity with the risk of reduced stego image quality.

Keywords: data hiding, reduced difference expansion, prediction error expansion, location map, interpolation.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirabbil'alamin. Segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tesis yang berjudul “Pengembangan Metode Reduced Difference Expansion” pada Data Hiding dengan Teknik Interpolasi Citra” dengan hasil yang memuaskan.

Pengerjaan Tesis ini merupakan suatu kesempatan yang sangat berharga bagi penulis, karena dalam proses pengerjaan Tesis ini penulis dapat memahami, meningkatkan, serta mengimplementasikan ilmu yang telah didapatkan penulis selama menempuh perkuliahan di Program Pasca Sarjana Teknik Informatika ITS.

Penyelesaian Buku Tesis ini, tidak terlepas dari bantuan dan dukungan semua pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Tesis ini dengan baik.
2. Orang tua penulis, Ibu Nuzula Mafruchah, S.AB, M.M., dan Bapak Mulyono, S.AB. yang telah memberikan motivasi serta doa tulus tak henti untuk penulis supaya mendapatkan hasil yang terbaik dalam setiap jenjang pendidikan yang ditempuh oleh penulis.
3. Fidyta Taufiqi Abdillah, S.IP., dan Sermatutar Arm. M. Hafizh Al Maliki, S.T.Han, selaku adik-adik kandung dari penulis yang sedang berjuang bersama-sama untuk menempuh proses penyelesaian tugas akhir di masing-masing instansi pendidikan yang ditempuh.
4. Bapak Tohari Ahmad, S.Kom, MIT., Ph.D, selaku dosen wali dan sebagai pembimbing yang telah memberikan kepercayaan, motivasi, bimbingan, nasihat, perhatian serta semua bantuan yang telah diberikan kepada penulis dalam menyelesaikan Tesis ini.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D., selaku dosen penguji dan Ketua Program Pasca Sarjana Teknik Informatika ITS.

6. Bapak Royyana Muslim I., S.Kom., M.Kom., Ph.D., dan Ibu Henning Titi, S.Kom., M.Kom., selaku dosen penguji yang telah memberikan saran, arahan, dan koreksi dalam pengerjaan Tesis ini.
7. Ibu Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. selaku Sekretaris Program Pasca Sarjana Teknik Informatika ITS yang telah memberikan perhatian khusus kepada penulis dalam menyelesaikan penelitian dan penyusunan buku.
8. Mbak Lina, Mas Kunto, dan segenap Staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kegiatan belajar mengajar di Teknik Informatika ITS.
9. Sahabat-sahabat penulis yang memberi dukungan mental, pengorbanan waktu dan energi khususnya Arief Budi Sucianto, S.I.Kom, yang telah berjuang bersama penulis dalam proses penyelesaian tugas akhir di bidang keahlian masing-masing. Mikan, Alfa, Inri, Yurike, Owen, Yuki, Toki, serta segenap rekan penulis yang tidak dapat disebutkan satu per satu di sini.
10. Rekan sejawat dalam Program Pasca Sarjana Teknik Informatika angkatan 2014 dan 2015 yang telah membantu penulis dalam proses penyelesaian Tesis.
11. Seluruh jajaran staf dan admin NIH yang telah menyediakan fasilitas untuk menulis sebagai selingan penulis dalam proses penyelesaian penulisan Buku Tesis.
12. Callum, Cyril, dan Alastair Carradine, Thyra Salisbury, Reginald Culpeper, dan Adrielle Winthorpe, selaku sahabat dan alter ego penulis. Tudor Knox, Romero Allenton, Lincoln Colton, Cicera dan Dionysius Carradine, Samantha Hays, Ethelstan Turing, Hiram Emsworth, Ishmael Sayler, dan Elisamarie Quinton sebagai alter ego dari rekan-rekan penulis yang selalu meringankan beban penulis saat proses penulisan Buku Tesis ini.

Penulis menyadari bahwa Buku Tesis ini masih jauh dari kata sempurna. Seluruh kritik dan saran dari pembaca akan diterima dengan pikiran terbuka oleh penulis. Akhir kata, waalaikumsalam warrahmatullahi wabarakatuh.

Surabaya, Januari 2017

Penulis

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Kontribusi Penelitian.....	4
1.7 Sistematika Penulisan	4
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Data Hiding	7
2.2 Difference Expansion.....	8
2.3 Prediction Error Expansion	11
2.4 Interpolasi pada Citra Digital	12
2.5 Parameter Pengujian.....	13
2.5.1 Peak Signal to Noise Ratio (PSNR).....	13
2.5.2 Bit-per-Pixel (BPP)	14
2.5.3 Structural Similarity (SSIM)	14
2.5.4 Universal Image Quality Index (UQI)	15
2.5.5 Kullback-Leibler (K-L) Divergence	16
BAB 3 METODE PENELITIAN.....	17
3.1 Tahapan Penelitian	17
3.2 Studi Literatur	17

3.3	Perancangan Algoritma	18
3.3.1.	Teknik Interpolasi.....	19
3.3.2.	Proses Penyisipan	23
3.3.3.	Proses Ekstraksi dan Pengembalian Citra Cover.....	27
3.3.4.	Location Map.....	32
3.4	Implementasi Penelitian	33
3.5	Perancangan Uji Coba dan Analisis Hasil.....	33
3.6	Penyusunan Buku Tesis.....	35
3.7	Jadwal Penelitian	35
BAB 4 PEMBAHASAN DAN ANALISA HASIL UJI COBA		37
4.1.	Pembahasan	37
4.1.1.	Proses Interpolasi.....	37
4.1.2.	Proses Penyisipan Tahap Pertama	43
4.1.3.	Proses Penyisipan Tahap Kedua.....	45
4.1.4.	Location Map.....	47
4.2.	Hasil Uji Coba dan Analisa	47
4.2.1.	Hasil Uji Coba Skenario 1	47
4.2.2.	Hasil Uji Coba Skenario 2	52
4.2.3.	Hasil Uji Coba Skenario 3	52
4.2.4.	Hasil Uji Coba Skenario 4	60
4.3.	Hasil Uji Coba Neighbor Mean Interpolation (NMI).....	61
4.4.	Hasil Uji Coba Directional Interpolation and Difference Expansion....	64
4.5.	Analisa Hasil Percobaan.....	66
BAB 5 KESIMPULAN DAN SARAN		69
5.1.	Kesimpulan.....	69
5.2.	Saran	70
DAFTAR PUSTAKA.....		71
BIODATA PENULIS.....		73

DAFTAR GAMBAR

Gambar 2.1 Pemetaan blok piksel pada PEE	12
Gambar 3.1 Diagram Alir Penelitian	17
Gambar 3.2 Diagram Alir Proses Penyisipan	18
Gambar 3.3 Sebuah blok piksel yang diinterpolasi.....	19
Gambar 3.4 Posisi blok piksel di pojok citra	20
Gambar 3.5 Komponen blok piksel yang berada di pojok citra.....	20
Gambar 3.6 Posisi blok piksel di sisi citra	21
Gambar 3.7 Komponen blok piksel yang berada di sisi citra	21
Gambar 3.8 Posisi blok piksel di tengah-tengah.....	22
Gambar 3.9 Komponen blok piksel yang berada di tengah citra	23
Gambar 3.10 Pemetaan blok piksel sebelum proses penyisipan.....	24
Gambar 3.11 Gambar piksel yang digunakan untuk menentukan nilai prediksi ..	25
Gambar 3.12 Pemetaan hasil akhir penyisipan pada blok piksel	27
Gambar 3.13 Diagram Alir Proses Ekstraksi	28
Gambar 4.1 Kode untuk Proses Interpolasi Pada Piksel Pojok Kiri Atas.....	37
Gambar 4.2 Kode untuk Proses Interpolasi Pada Piksel Pojok Kanan Atas.....	38
Gambar 4.3 Kode untuk Proses Interpolasi Pada Piksel Pojok Kanan Bawah	39
Gambar 4.4 Kode untuk Proses Interpolasi Pada Piksel Pojok Kiri Bawah	39
Gambar 4.5 Kode untuk Proses Interpolasi Pada Piksel Sisi Atas.....	40
Gambar 4.6 Kode untuk Proses Interpolasi Pada Piksel Sisi Kiri	40
Gambar 4.7 Kode untuk Proses Interpolasi Pada Piksel Sisi Kanan	41
Gambar 4.8 Kode untuk Proses Interpolasi Pada Piksel Sisi Bawah.....	42
Gambar 4.9 Kode untuk Proses Interpolasi Pada Piksel Tengah	42
Gambar 4.10 Posisi Piksel dalam Sebuah Blok Piksel	43
Gambar 4.11 Kode untuk Mencari Selisih antara Piksel Asli dan Piksel Hasil Interpolasi Citra.....	44
Gambar 4.12 Kode untuk Penyisipan Tahap Pertama	45
Gambar 4.11 Kode untuk Mencari Selisih antara Piksel Referensi dengan Piksel Hasil Interpolasi	46

Gambar 4.12 Kode Penyisipan Tahap Kedua.....	46
Gambar 4.13 Kode untuk Menentukan Location Map	47
Gambar 4.14 Grafik Perbandingan Kapasitas dan Kualitas Citra Lena	48
Gambar 4.15 Grafik Perbandingan Kapasitas dan Kualitas Citra Plane	49
Gambar 4.16 Grafik Perbandingan Kapasitas dan Kualitas Citra Pepper	50
Gambar 4.17 Grafik Perbandingan Kapasitas dan Kualitas Citra Baboon.....	51
Gambar 4.18 Penyisipan pada Citra Boat.....	52
Gambar 4.19 Posisi blok piksel di pojok citra.....	53
Gambar 4.20 Komponen blok piksel yang berada di pojok citra	53
Gambar 4.21 Grafik Perubahan Pembobotan Sisi Pojok	54
Gambar 4.22 Posisi blok piksel di sisi citra	55
Gambar 4.23 Komponen blok piksel yang berada di sisi citra.....	55
Gambar 4.24 Grafik Perubahan Pembobotan Sisi Luar.....	56
Gambar 4.25 Grafik Perubahan Pembobotan Sisi Sisi Tepi Dalam	58
Gambar 4.26 Posisi blok piksel di tengah-tengah	58
Gambar 4.27 Komponen blok piksel yang berada di tengah citra	58
Gambar 4.28 Grafik Perubahan Pembobotan Sisi Tengah	60
Gambar 4.29 Grafik Perbandingan dengan Metode NMI Citra Lena	62
Gambar 4.30 Grafik Perbandingan dengan Metode NMI Citra Plane	62
Gambar 4.31 Grafik Perbandingan dengan Metode NMI Citra Pepper	63
Gambar 4.32 Grafik Perbandingan dengan Metode NMI Citra Baboon.....	63
Gambar 4.33 Grafik Perbandingan dengan Metode NMI Citra Boat.....	64
Gambar 4.34 Grafik Perbandingan Metode DI Citra Lena	65
Gambar 4.35 Grafik Perbandingan Metode DI Citra Plane.....	65
Gambar 4.36 Grafik Perbandingan Metode DI Citra Boat	66

DAFTAR TABEL

Tabel 3.1 Model Location Map.....	32
Tabel 3.2 Jadwal Pelaksanaan Penelitian.....	35
Tabel 4.1 Hasil Penyisipan pada Citra Lena	48
Tabel 4.2 Hasil Penyisipan pada Citra Plane	49
Tabel 4.3 Hasil Penyisipan pada Citra Pepper	49
Tabel 4.4 Hasil Penyisipan pada Citra Baboon.....	50
Tabel 4.5 Hasil Penyisipan pada Citra Boat.....	51
Tabel 4.6 Hasil Pengujian Steganalysis pada Masing-masing Citra.....	52
Tabel 4.7 Uji Coba Nilai Pembobotan pada Sisi Pojok	53
Tabel 4.8 Hasil PSNR dari Pengubahan Pembobotan Sisi Pojok	54
Tabel 4.9 Uji Coba Nilai Pembobotan pada Sisi Luar	55
Tabel 4.10 Hasil PSNR dari Pengubahan Pembobotan Sisi Luar	56
Tabel 4.11 Uji Coba Nilai Pembobotan pada Sisi Tepi Dalam	57
Tabel 4.12 Hasil PSNR dari Pengubahan Pembobotan Sisi Tepi Dalam	57
Tabel 4.13 Uji Coba Nilai Pembobotan pada Sisi Tengah	59
Tabel 4.14 Hasil PSNR dari Pengubahan Pembobotan Sisi Tengah	59
Tabel 4.15 Hasil Uji Coba Penggantian Nilai Thresholding	60
Tabel 4.16 Hasil Uji Coba Penyisipan Pesan dengan NMI	61
Tabel 4.17 Hasil Uji Coba Directional Interpolation dan DE.....	64

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini penggunaan Internet sebagai media untuk bertukar informasi menjadi hal yang lazim dilakukan. Sebagai salah satu jalur komunikasi utama yang kerap digunakan, maka perlindungan terhadap data yang ditransmisikan melalui jaringan internet harus digiatkan. Salah satu usaha untuk melindungi data yang sensitif adalah dengan teknik steganografi. Steganografi adalah sebuah teknik menyembunyikan data penting ke dalam sebuah data lain yang digunakan sebagai media pengiriman. Hal ini dilakukan sebagai kamuflase dari pihak-pihak yang tidak memiliki otorisasi untuk melihat data tersebut (Husrev T. Sencar, 2004).

Penelitian teknik *data hiding* masih kerap dilakukan. Salah satu alasannya adalah untuk mengembangkan berbagai metode yang telah ada sebagai upaya untuk meningkatkan hasilnya. Dengan semakin berkembangnya jaman, maka jumlah data yang harus dapat disimpan dalam sebuah citra stego juga semakin besar. Namun hal ini diharapkan supaya tidak mengurangi kualitas citra stego yang digunakan. Di sisi lain, apabila sebuah gambar memiliki ukuran yang terlampau besar sehingga dianggap tidak lazim, maka citra stego tersebut dapat dicurigai sebagai sebuah media penyimpanan data rahasia. Hal ini dapat dianggap sebagai kegagalan dalam penggunaan teknik steganografi.

Saat ini metode *data hiding* yang banyak dikembangkan adalah *Difference Expansion* (DE). Metode DE ini menggunakan selisih antara dua piksel yang bertetangga untuk menyembunyikan bit data (Tian, 2003). Karena kemudahannya, metode ini dikembangkan menjadi berbagai macam metode baru turunan DE yang dapat meningkatkan kapasitas dari jumlah pesan rahasia yang dapat disimpan. Kekurangan dari metode ini adalah diperlukan adanya sebuah *location map* untuk menentukan posisi bit data rahasia yang ada dalam sebuah piksel tertentu. Hal ini dikarenakan tidak semua piksel dapat menyimpan bit data dari pesan rahasia karena nilainya yang terlalu tinggi atau terlalu rendah.

Metode lain yang dikembangkan dari metode DE adalah *Prediction Error Expansion* (PEE). PEE dikembangkan karena tuntutan penambahan kapasitas pesan yang dapat disimpan pada sebuah citra stego. Metode ini menyimpan bit data pesan rahasia berdasarkan nilai perkiraan dari ketiga piksel yang bertetangga. Kemudian menempelkan bit pesan tersebut pada nilai piksel yang baru setelah ditambahkan dengan nilai prediksi. PEE tidak membutuhkan *location map* karena proses penyimpanan pesan dilakukan secara sekuensial (Diljith & Jeffrey, 2007). Namun kekurangan dari metode ini adalah menurunnya kualitas citra stego.

Penelitian yang telah dilakukan untuk meningkatkan kedua variabel dalam citra steganografi (kapasitas dan kualitas) adalah dengan menggunakan teknik interpolasi citra. Sebuah citra diinterpolasi dengan ukuran tetap untuk menghasilkan nilai prediksi piksel baru, kemudian dilakukan penyimpanan pesan ke dalam piksel baru berdasarkan hasil dari proses interpolasi untuk menciptakan sebuah nilai prediksi yang tidak memiliki perbedaan begitu jauh dengan piksel-piksel yang bertetangga (Govind, 2015).

Penggabungan kedua metode ini dapat menghilangkan penggunaan *location map* dan menjaga kualitas citra stego yang digunakan. Dalam beberapa kasus penggunaan citra medis dan peta yang digunakan sebagai citra stego terjadinya distorsi tidak dapat ditoleransi. Metode yang digunakan dalam penelitian sebelumnya adalah *Neighbor Mean Interpolation* (NMI) dengan menggunakan rata-rata dari nilai piksel yang saling bertetangga sebagai nilai basis (Jung, 2009).

Sedangkan pada penelitian sebelumnya penggunaan metode interpolasi yang konvensional memperhitungkan kemudahan dari perhitungan nilai perkiraan piksel-piksel yang baru pada perluasan satu buah blok piksel. Metode *Nearest Neighbor* dapat menghasilkan nilai piksel baru yang nilainya tidak berselisih jauh dengan piksel lain yang bertetangga pada satu blok. Hal ini dapat menghasilkan hasil citra yang terinterpolasi dengan kompleksitas perhitungan yang rendah namun tidak membutuhkan proses klasifikasi karena tidak mengubah nilai *grey level* sebuah citra (Jung, 2009).

Penelitian ini mengusulkan metode *data hiding* yang digabungkan dengan teknik interpolasi pada citra digital untuk menambahkan kapasitas data rahasia yang disimpan. Dengan menggunakan sebuah citra digital yang dibagi menjadi blok

piksel tidak tumpang tindih berukuran 3×3 piksel, kemudian dilakukan interpolasi untuk mendapatkan prediksi nilai baru piksel. Pesan rahasia dimasukkan ke dalam citra yang baru menggunakan dua kali proses penyisipan. Proses yang pertama dilakukan penyisipan pada piksel yang nilai barunya berasal dari proses interpolasi, sedangkan proses penyisipan yang kedua dilakukan pada piksel referensi yang tidak diinterpolasi. Diharapkan penelitian ini menghasilkan sebuah metode *data hiding* yang dapat meningkatkan kapasitas data dalam sebuah citra namun dengan resiko sedikit penurunan kualitas yang mungkin terjadi.

1.2 Perumusan Masalah

Dari latar belakang diatas didapatkan permasalahan sebagai berikut:

1. Bagaimana melakukan teknik interpolasi pada citra digital berdasarkan piksel yang saling bertetangga?
2. Bagaimana melakukan proses penyisipan pada piksel berdasarkan nilai prediksi yang merupakan hasil dari teknik interpolasi?
3. Bagaimana melakukan proses penyisipan pada piksel referensi yang tidak terpengaruh oleh teknik interpolasi?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam penelitian ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. *Cover media* yang digunakan adalah berupa citra digital.
2. Data citra digital yang digunakan berupa data set citra testing standar *greyscale* dengan ukuran 512×512 piksel (Carolina, 2015).
3. Pesan rahasia yang disisipkan berupa teks pengisi standar penerbitan "*Lorem ipsum*".

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

1. Melakukan proses penyisipan informasi dengan mengintegrasikan teknik interpolasi dan steganografi pada sebuah citra digital.
2. Mendapatkan pesan rahasia yang disisipkan dan mengembalikan citra dari hasil ekstraksi pada citra stego.
3. Mengembangkan sebuah metode modifikasi DE yang meminimalisasi jumlah piksel yang dijadikan *base point*.
4. Menguji metode yang diusulkan untuk mengetahui jumlah kapasitas maksimal pada setiap citra stego standar.

1.5 Manfaat Penelitian

Dengan menggunakan metode ini diharapkan mampu meningkatkan kapasitas dari citra stego yang digunakan pada citra sehingga kecurigaan terhadap sebuah citra stego dapat dihindari.

1.6 Kontribusi Penelitian

Kontribusi pada penelitian ini adalah mengembangkan metode penyembunyian data dengan mengoptimalkan penggunaan piksel hasil interpolasi guna meningkatkan kapasitas penyembunyian data dengan risiko penurunan kualitas citra stego dan menambah kompleksitas proses penyisipan pesan.

1.7 Sistematika Penulisan

Sistematika penulisan pada penelitian ditujukan untuk memberikan gambaran dan uraian dari penelitian. Secara garis besar terdapat beberapa bab, sebagai berikut:

BAB 1: PENDAHULUAN

Memaparkan tentang latar belakang masalah yang kemudian dirumuskan ke dalam rumusan masalah dalam bentuk uraian terstruktur dan dilengkapi dengan batasan masalah, tujuan penelitian, manfaat penelitian, kontribusi serta sistematika penulisan

BAB 2: KAJIAN PUSTAKA DAN DAFTAR TEORI

Menjelaskan tentang metode yang digunakan dan teori-teori pendukung dalam penelitian.

BAB 3: METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan mengumpulkan data dan permasalahan. Data selanjutnya diolah dengan menggunakan metode yang diajukan untuk mendapatkan pengetahuan serta informasi yang bisa dimanfaatkan oleh pengguna.

BAB 4: HASIL DAN PEMBAHASAN

Bab ini menjabarkan dan menjelaskan hasil yang diperoleh dari hasil penelitian berupa analisa terhadap pengetahuan yang dihasilkan dari pengguna menggunakan metode yang diusulkan.

BAB 5: KESIMPULAN DAN SARAN

Menguraikan kesimpulan yang diambil berdasarkan hasil dari penelitian yang telah diperoleh.

[Halaman ini sengaja dikosongkan]

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dibahas beberapa teori dasar yang menunjang dalam penelitian ini.

2.1 *Data Hiding*

Penyembunyian data merupakan sebuah seni menyembunyikan sebuah sinyal pesan dalam sebuah sinyal induk tanpa terjadinya distorsi sinyal (Husrev T. Sencar, 2004). Salah satu jenis penyembunyian data adalah steganografi. Secara hafiah, steganos artinya tertutup, sedangkan graphia artinya menulis. Sehingga steganografi adalah seni berkomunikasi secara tersembunyi.

Dewasa ini, steganografi adalah sebuah teknik penyembunyian sebuah data baik berupa pesan, citra maupun video ke dalam data lain yang berupa pesan, citra atau video. Steganografi sering disejajarkan penggunaannya dengan kriptografi, dengan perbedaan utama pada bagian penyembunyian data (*data hiding*) pada teknik steganografi untuk meniadakan timbulnya kecurigaan. Sedangkan pada kriptografi pesan yang terbentuk dalam kode dapat memancing perhatian orang-orang yang tidak memiliki kepentingan (Tian, 2003).

Terdapat berbagai macam metode yang awalnya hanya dapat digunakan untuk menyembunyikan pesan rahasia dan mengambil pesan tersebut kembali. Sayangnya, untuk mengembalikan kualitas asli media tersebut tidak memungkinkan (*irreversible*). Sehingga berkembanglah metode steganografi yang dapat mengembalikan kualitas orisinal dari citra stego yang telah digunakan (*reversible*). Pada tahun 1997, metode steganografi yang reversible adalah Barton Patent. Metode ini bit data yang akan disembunyikan ke data lain dikompres kemudian di-*embed* pada blok data (Barton, 1997). Sedangkan pada penelitian milik (Honsinger, 2001) merekonstruksi payload dari sebuah citra stego, kemudian mengurangi payload tersebut dari citra stego untuk mengembalikan citra orisinal. (Macq, 2000) melebarkan algoritma *patchwork* untuk mendapatkan *data-*

embedding yang *reversible*. (Fridrich, 2002) mengembangkan sebuah teknik *data-embedding* yang *reversible* dan berkapasitas tinggi berdasarkan pesan yang harus disimpan pada bit yang berada dalam sebuah kelompok piksel. Mereka juga menjelaskan dua teknik *data-embedding* yang *reversible* untuk citra berformat JPG. Sedangkan (De Vleeschouwer, 2003) mengembangkan sebuah algoritma *data-embedding* berdasarkan interpretasi sirkular dari transformasi *one-to-one* (*bijective*). (Kalker, 2003) menyatakan secara teoritis batasan kapasitas dari kompresi data lossless berdasarkan *data-embedding* yang *reversible*. (Celik, 2002) menunjukkan sebuah algoritma *data-embedding* yang memiliki kapasitas tinggi dan distorsi rendah dengan mengompres sisa hasil kuantisasi.

Dewasa ini penggunaan metode steganografi *irreversible* sudah jarang digunakan karena ketidakmampuannya dalam mengembalikan kualitas asli citra orisinal. Penggunaan metode steganografi *reversible* kini lebih sering digunakan, seperti metode *Difference Expansion* (DE) dan turunannya yang akan dibahas pada subbab selanjutnya.

2.2 *Difference Expansion*

Metode *Difference Expansion* adalah salah satu contoh dari metode *reversible data-embedding* atau *lossless data-embedding*. Metode ini melakukan proses *embedding* data rahasia yang kemudian disebut *payload* ke sebuah citra digital yang dapat dikembalikan seperti semula (*reversible*). Syarat utama dari metode *reversible data-embedding* ini adalah penurunan kualitas citra harus sangat rendah.

Metode ini diharapkan dapat meminimalisasi distorsi pada citra stego yang digunakan. Apabila citra stego yang digunakan untuk kepentingan militer dan medis, maka perubahan piksel sekecil apa pun dapat mengubah persepsi dan pemahaman pihak lain. Maka dari itu, dibutuhkan sebuah metode yang dapat mengembalikan kualitas citra stego seperti aslinya.

Apabila ditelisik lebih jauh, maka metode *reversible data-embedding* juga dapat digunakan sebagai pembawa informasi. Karena selisih nilai piksel yang berubah tidak dapat ditangkap oleh mata telanjang manusia, maka penggunaannya

dapat membantu pengiriman pesan rahasia tanpa perlu adanya sebuah *metadata* yang ditempelkan (Tian, 2003).

Metode DE menyisipkan *payload* ke dalam LSB dari selisih kedua piksel yang bertetangga sebelum akhirnya nilai tersebut dimasukkan kembali ke kedua piksel. Dengan beberapa perbedaan pada operasi pada kedua piksel, sehingga kedua piksel yang bertetangga tersebut memiliki selisih yang cukup jauh. Apabila u_1 dan u_2 adalah sepasang piksel yang bertetangga, maka langkah pertama adalah mencari nilai rata-rata dan selisih dari kedua piksel tersebut, seperti pada Persamaan 2.1.

$$l = \left\lfloor \frac{u_1 + u_2}{2} \right\rfloor, \quad h = u_1 - u_2 \quad (2.1)$$

Kemudian pesan rahasia atau *payload* b disisipkan dalam citra dengan memperlebar selisih antara kedua piksel bertetangga tersebut seperti pada Persamaan 2.2.

$$h' = 2 \times h + b \quad (2.2)$$

Setelah selisih baru didapatkan, maka berikutnya dilakukan penghitungan nilai piksel baru yang akan digunakan untuk membentuk citra stego seperti pada Persamaan 2.3.

$$x' = l + \left\lfloor \frac{h' + 1}{2} \right\rfloor, \quad y' = l - \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.3)$$

Proses ekstraksi pesan (b) dari nilai piksel baru (x', y'), akan menghasilkan nilai piksel yang sebenarnya, seperti pada Persamaan 2.4 dan 2.5.

$$l' = \left\lfloor \frac{x' + y'}{2} \right\rfloor, \quad h' = x' - y' \quad (2.4)$$

$$b = LSB(h'), \quad h = \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.5)$$

Setelah mendapatkan nilai h' maka perlu diubah menjadi bilangan berbasis biner untuk mengambil LSB yang merupakan pesan (b) dan selisih asli (h) sehingga nilai piksel asli (x , y) dapat dikembalikan.

Metode *Generalized Difference Expansion* (GDE) yang diajukan oleh Alattar menggunakan sebuah piksel sebagai *base point* (Alattar, 2004). Kemudian menghitung selisih antara piksel yang digunakan sebagai *base point* dengan piksel lain yang bertetangga. Terdapat dua jenis proses *embedding* pada metode ini: *expandable* dan *changeable*.

Proses *embedding* secara *expandable* mirip dengan metode utama DE. Namun apabila terjadi keadaan *overflow* atau *underflow* maka proses *embedding* menggunakan teknik *changeable*. Keadaan *overflow* adalah keadaan dimana nilai piksel baru setelah disisipi oleh pesan rahasia akan melebihi 255. Sedangkan keadaan *underflow* adalah keadaan ketika nilai piksel baru setelah disisipi pesan rahasia akan kurang dari nol.

Metode *Reduced Difference Expansion* (RDE) berawal dari semakin gencarnya penggunaan internet untuk kepentingan medis. Hal ini dilakukan sebagai alternatif penyimpanan data-data sensitif dalam citra medis. Penyembunyian data pasien yang bersangkutan dan *copyright* disimpan menggunakan metode RDE dalam sebuah citra pada lapisan-lapisan domain spasial (Lou, 2009).

Sebuah teknik *data hiding* harus memiliki empat syarat dasar berikut ini: *imperceptibility* (sulit untuk dipahami oleh manusia), *robustness* (ketahanan), kapasitas, dan keamanan. Namun bukanlah hal yang mudah untuk memenuhi keempat syarat tersebut. Misalnya melakukan proses *embedding* dengan jumlah pesan yang begitu banyak dapat menyebabkan citra tersebut mudah terdeteksi. Di sisi lain, untuk meningkatkan tingkat *imperceptibility*, dapat menyebabkan kapasitas yang untuk menyimpan pesan rahasia berkurang. Maka dari itu pengembangan RDE ini dilakukan untuk memenuhi empat syarat mendasar tersebut.

Dengan mengalikan dua selisih antara dua piksel yang bertetangga, hal ini menyebabkan nilai kedua piksel baru tersebut sangat jauh. Dalam RDE hal ini dihindari dengan cara mengurangi nilai selisih apabila memiliki nilai lebih dari dua seperti pada Persamaan 2.6.

$$h' = \begin{cases} h, & \text{jika } h < 2 \\ h - 2^{\lfloor \log_2 h \rfloor - 1}, & \text{jika } h \geq 2 \end{cases} \quad (2.6)$$

Sedangkan pada tahun 2014, metode baru yang mengusulkan integrasi dari metode GDE dan RDE (Ahmad, 2014). Metode ini melakukan pemilihan base point dari GDE yang mengambil salah satu piksel diubah menjadi nilai median dari sebuah blok piksel yang digunakan untuk menyisipkan data. Lantas dilakukan reduksi selisih apabila nilai selisih yang dimiliki lebih dari dua dan kurang dari minus dua seperti pada Persamaan 2.7.

$$\begin{cases} h, & \text{jika } -2 < h < 2 \\ h + 2^{\lfloor \log h \rfloor - 1}, & \text{jika } -2 \leq h \\ h - 2^{\lfloor \log h \rfloor - 1}, & \text{jika } h \geq 2 \end{cases} \quad (2.7)$$

Penyisipan pada metode ini dilakukan seperti pada GDE, menggunakan RDE yang expendable, atau expendable namun bukan RDE, changeable, dan unchangeable. Penyisipan hanya dilakukan pada tiga kategori awal, sedangkan kategori unchangeable tidak bisa dilakukan penyisipan.

2.3 Prediction Error Expansion

Metode DE yang diajukan oleh (Tian, 2003) memiliki kapasitas pesan rahasia yang banyak, namun sayangnya metode ini menyebabkan distorsi pada kapasitas *embedding* yang kecil dan kurangnya kontrol untuk kapasitas karena membutuhkan *location map*. Metode yang diusulkan oleh (Diljith & Jeffrey, 2007) ini lebih mengeksplorasi korelasi dari nilai piksel yang bertetangga daripada metode DE. Penggabungan pergeseran histogram (*histogram shifting*) dan PEE dapat menghasilkan metode yang efektif untuk *data embedding*. PEE dapat melipatgandakan jumlah kapasitas yang dapat disisipkan dua kali lebih banyak daripada metode DE.

Pada metode PEE, tidak memerlukan *location map* sehingga proses *embedding*-nya memiliki urutan yang jelas. Seperti pada Gambar 2.1, piksel *a*

memiliki tiga konteks atau piksel referensi yaitu c_1 , c_2 , dan c_3 . Nilai baru piksel a ditentukan oleh Persamaan 2.8.

c_1	c_2
c_2	a

Gambar 2.1 Pemetaan blok piksel pada PEE

$$\tilde{a} = \begin{cases} \max(c_2, c_3), & \text{if } c_1 \leq \min(c_2, c_3) \\ \min(c_2, c_3), & \text{if } c_1 \geq \max(c_2, c_3) \\ c_2 + c_3 - c_1, & \text{otherwise} \end{cases} \quad (2.8)$$

Nilai prediksi piksel a ditetapkan pada nilai integer genap yang lebih dari atau sama dengan a , seperti pada Persamaan 2.9.

$$\hat{a} = 2\lfloor \tilde{a}/2 \rfloor \quad (2.9)$$

2.4 Interpolasi pada Citra Digital

Dewasa ini penggunaan media digital semakin marak sehingga kemampuan untuk menginterpolasi citra secara digital menjadi sebuah citra dengan kualitas yang lebih baik dibutuhkan implementasinya. Beberapa faktor penting yang dipertimbangkan dalam memilih sebuah metode interpolasi adalah ketajaman tepi citra, tidak terpengaruh dengan artifak, dan kemudahan dalam proses penghitungan.

Salah satu metode interpolasi yang paling mudah adalah *Nearest Neighbor Interpolation* (NNI) yang melakukan resampling pada sebuah blok piksel. Metode *Nearest Neighbor* dapat menghasilkan nilai piksel baru yang nilainya tidak berselisih jauh dengan piksel lain yang bertetangga pada satu blok. Hal ini dapat menghasilkan hasil citra yang terinterpolasi dengan kompleksitas perhitungan yang rendah namun tidak membutuhkan proses klasifikasi karena tidak mengubah nilai *grey level* sebuah citra (Jung, 2009).

2.5 Parameter Pengujian

Beberapa parameter pengujian yang akan digunakan pada penelitian ini akan dijelaskan dalam subbab ini.

2.5.1 *Peak Signal to Noise Ratio (PSNR)*

Steganografi yang menggunakan media citra digital sebaiknya memiliki distorsi seminimal mungkin dan artifak-artifak yang membuatnya dicurigai sebagai media stego. Karena itulah, citra orisinil dan citra stego alangkah baiknya memiliki kemiripan sebanyak mungkin dalam kondisi apa pun.

Untuk mengevaluasi kualitas citra dari banyak jenis algoritma kompresi citra digital (misalnya kompresi *lossy* dan steganografi pada citra) memiliki banyak halangan seperti, jumlah degradasi yang disebabkan oleh citra yang telah direkonstruksi. Terdapat dua cara untuk mengukur kualitas citra digital, yaitu secara obyektif dan subyektif. Metode obyektif menghitung aspek-aspek fisik dari citra, sedangkan metode subyektif menggunakan indra manusia untuk mengamati perubahan yang terjadi pada citra stego dan citra orisinil.

$$MSE = \left(\frac{1}{MN} \right) \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \overline{X}_{ij})^2 \quad (2.10)$$

$$PSNR = 10 \cdot \log_{10} \frac{I^2}{MSE} \quad db \quad (2.11)$$

Dalam Persamaan 2.10 merupakan *Mean Square Error*, dengan masing-masing komponen persamaannya adalah, M dan N merupakan dimensi dari dua citra yang dibandingkan. Sedangkan i dan j merupakan koordinat dari citra yang dibandingkan. X_{ij} merupakan nilai piksel citra stego pada koordinat tersebut, sedangkan \overline{X}_{ij} merupakan nilai piksel citra cover.

Sedangkan dalam rumus penghitungan PSNR dalam Persamaan 2.11 dinyatakan dalam skala logaritmik dengan satuan decibel (dB). Dengan I^2 merupakan nilai kemungkinan maksimum dari sebuah citra. Apabila piksel direpresentasikan dengan 8 bit, maka nilainya adalah 255.

PSNR menghitung kemiripan antara dua citra dan seberapa mirip kedua citra itu antar satu sama lain. Sedangkan MSE menghitung perbedaan yang ada di

antara dua citra tersebut. Karena kompleksitas yang rendah dan waktu penghitungan yang relatif sedikit maka kedua metrik ini kerap digunakan (Almohammad, 2010).

Sehingga hasil citra stego yang lebih baik adalah yang memiliki nilai MSE sangat rendah atau hampir mendekati nol karena hal tersebut berarti bahwa perbedaan antara citra orisinal dengan citra stego dapat diabaikan. Oleh sebab itu, penggunaan metode evaluasi kualitas citra secara obyektif akan lebih cepat dan efektif dari segi biaya daripada secara subyektif.

2.5.2 *Bit-per-Pixel (BPP)*

Pengukuran kapasitas pesan rahasia maksimal yang dapat ditampung oleh sebuah citra dihitung dengan metrik *Bit-per-Pixel* (BPP). Dengan menjumlah seluruh bit total yang disembunyikan dibagi dengan luas piksel citra yang digunakan akan mendapatkan sebuah metrik yang digunakan untuk mengukur kapasitas sebuah metode steganografi.

$$BPP = \frac{\text{Jumlah total pesan}}{\text{Jumlah piksel sebuah citra}} \quad (2.12)$$

2.5.3 *Structural Similarity (SSIM)*

Structural similarity merupakan sebuah metrik yang digunakan untuk mengukur kemiripan antar dua gambar. Perbandingan dilakukan pada citra orisinal hasil interpolasi dengan citra stego yang telah memiliki pesan rahasia. SSIM menggunakan iluminasi, kontras, dan struktur citra.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.13)$$

$$C_1 = (K_1L)^2, \quad C_2 = (K_2L)^2 \quad (2.14)$$

Pada Persamaan 2.13 merupakan sebuah persamaan yang menggabungkan tiga komponen utama dari perhitungan SSIM, yaitu perbandingan iluminasi, kemudian kontras antara dua titik yang sama dalam dua citra, kemudian struktur antara dua citra tersebut. Pada persamaan di atas, μ merupakan rata-rata, σ adalah

varian, sedangkan C_1 dan C_2 adalah variabel untuk menstabilkan pembagi kecil seperti pada Persamaan 2.14. Untuk nilai $K_1 = 0,001$, $K_2 = 0,003$ merupakan nilai standar, dan L adalah *dynamic range* dari nilai piksel pada posisi tersebut (Wang, 2004).

2.5.4 Universal Image Quality Index (UQI)

Pengukuran kualitas citra secara obyektif menggunakan *Universal Quality Index* (UQI) dianggap memiliki kelebihan tersendiri. Pertama, proses penghitungan yang cukup mudah karena kompleksitasnya yang rendah. Kedua, indeks ini tidak bergantung pada kondisi saat melihat citra tersebut. Secara matematis indeks ini akan menghitung: terdapatnya korelasi, distorsi pencahayaan, dan distorsi kontras. UQI merupakan metode pengukuran citra pendahulu dari metode SSIM.

$$Q = \frac{4\sigma_{xy}\mu_x\mu_y}{(\sigma_x^2 + \sigma_y^2)[(\mu_x^2 + \mu_y^2)]} \quad (2.15)$$

Pada Persamaan 2.15, x dan y merupakan dua citra yang dibandingkan. Menggunakan nilai total seluruh piksel dari kedua citra, varian dari kedua koordinat citra, serta kovarian dari kedua citra. Ditunjukkan lebih detail pada Persamaan 2.16 hingga 2.20. (Zhou Wang, 2002)

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.16)$$

$$\mu_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.17)$$

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \quad (2.18)$$

$$\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2 \quad (2.19)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (2.20)$$

2.5.5 Kullback-Leibler (K-L) Divergence

KL Divergence merupakan salah satu parameter yang populer untuk menganalisa sebuah sistem steganografi. Misalkan X dan Y adalah citra *cover* dan citra stego, sedangkan p_x dan q_y adalah fungsi pendistribusian probabilitas dari X dan Y . Maka *KL Divergence* dari kedua fungsi pendistribusian probabilitas adalah sebagai berikut:

$$D(p_x \| q_y) = \sum_{g \in G} p_x(g) \log \frac{p_x(g)}{q_y(g)} \quad (2.21)$$

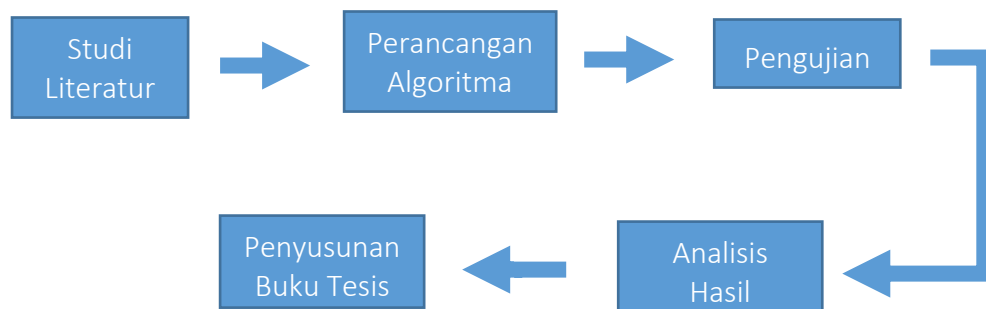
Dengan nilai $g \in G = \{0, 1, 2, \dots, 255\}$, merupakan nilai piksel pada media citra *grayscale*. Dapat disimpulkan bahwa fungsi persamaan ini adalah menghitung perkiraan dari perbedaan logaritmik antara probabilitas P dan Q . Apabila parameter korelasi p semakin besar, maka nilai *KL Divergence* juga semakin besar sehingga kemungkinan data rahasia untuk terungkap juga semakin besar.

BAB 3

METODE PENELITIAN

3.1 Tahapan Penelitian

Sebelum memulai penelitian ini, untuk mencapai tujuan yang diharapkan maka diperlukan beberapa langkah. Seperti yang dijelaskan pada Gambar 3.1, penelitian ini diawali dengan proses studi literatur untuk mempelajari permasalahan dan penelitian yang terkini. Setelah mempelajarinya, maka perancangan algoritma yang diusulkan yaitu penggabungan metode steganografi yang digabungkan dengan teknik interpolasi pada citra digital. Kemudian proses pengujian dilakukan pada metode yang diusulkan. Setelah itu analisis hasil pengujian akan dilakukan. Terakhir, semua proses dan hasil penelitian akan disusun dalam buku tesis.



Gambar 3.1 Diagram Alir Penelitian

3.2 Studi Literatur

Studi literatur dalam suatu penelitian dilakukan untuk membantu proses analisa awal sebuah permasalahan. Hal ini berfungsi untuk menguatkan pemahaman dari sisi konsep atau dasar teori yang akan digunakan. Dalam proses ini akan dipelajari semua informasi dan sumber pustaka yang disesuaikan dengan konteks penelitian. Termasuk di dalamnya mempelajari penelitian-penelitian terkait yang harus dipahami sebelum melakukan perancangan algoritma.

Dalam tahapan studi literature ini dilakukan eksplorasi putaka, *paper*, materi, dan informasi yang terkait dengan penelitian yang akan dilakukan.

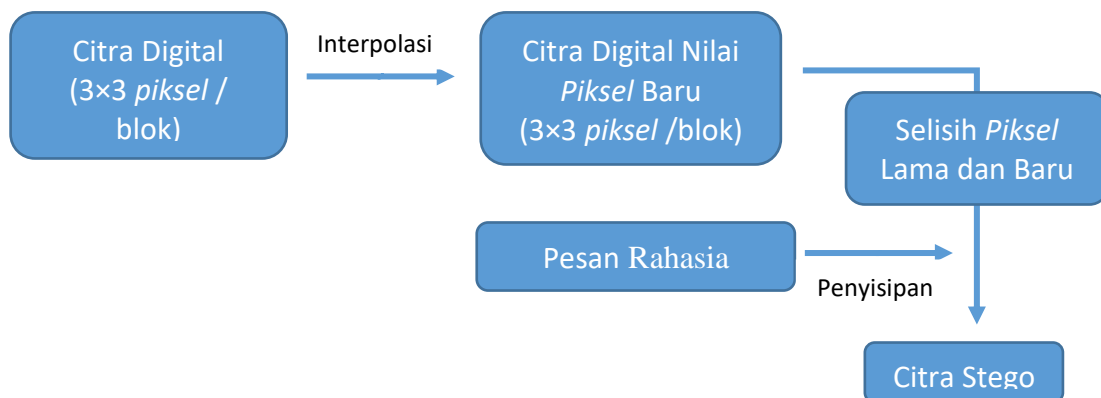
Beberapa topik literatur yang perlu dipelajari pada penelitian ini adalah sebagai berikut:

1. Konsep dasar penyembunyian data
2. Metode penyembunyian data *Difference Expansion* (DE)
3. Metode penyembunyian data *Prediction Error Expansion* (PEE)
4. Teknik interpolasi citra digital.

3.3 Perancangan Algoritma

Penggunaan metode DE dalam penyembunyian data pada proses steganografi menyebabkan nilai piksel yang baru memiliki rentang yang cukup jauh namun dengan kapasitas yang cukup rendah. Sedangkan penggunaan metode PEE pada proses penyembunyian data dapat menyebabkan penurunan kualitas karena penggeseran histogram (*histogram shifting*) yang dilakukannya.

Salah satu teknik yang dapat mengkombinasikan kedua metode ini adalah dengan teknik interpolasi. Teknik interpolasi pada citra digital dilakukan supaya penyisipan dapat dilakukan pada piksel baru dari hasil interpolasi. Hal ini dilakukan untuk menambah kapasitas penyisipan menggunakan metode yang diusulkan.



Gambar 3.2 Diagram Alir Proses Penyisipan

3.3.1. Teknik Interpolasi

Sebelum melakukan proses penyisipan pada citra digital, yang perlu diketahui adalah proses keseluruhan dari metode yang diajukan. Sebuah citra yang akan digunakan sebagai media stego dibagi menjadi blok piksel berukuran 3×3 yang tidak saling beririsan.



Gambar 3.3 Sebuah blok piksel yang diinterpolasi

Teknik interpolasi yang digunakan pada penelitian sebelumnya adalah *Neighbor Mean Interpolation*. Nilai piksel baru dibentuk berdasarkan rata-rata dari nilai piksel baru yang saling bertetangga. Perhitungan piksel baru yang diinterpolasi ditampilkan pada Persamaan 3.1 hingga Persamaan 3.4 (Govind, 2015).

$$IP_1 = \frac{RP_1 + RP_2}{2} \quad (3.1)$$

$$IP_2 = \frac{RP_1 + RP_3}{2} \quad (3.2)$$

$$IP_3 = \frac{RP_2 + RP_4}{2} \quad (3.3)$$

$$IP_4 = \frac{RP_3 + RP_4}{2} \quad (3.4)$$

Dengan :

IP adalah *interpolated pixel*.

RP adalah *reference pixel*.

Sedangkan teknik interpolasi yang diusulkan menggunakan pembobotan dari piksel-piksel yang bertetangga meskipun tidak dalam satu blok yang sama. Namun terdapat perbedaan pembobotan dalam beberapa posisi piksel yang disebabkan oleh posisinya, tidak memiliki jumlah piksel yang sama. Sehingga teknik interpolasi yang diajukan ini dibagi menjadi tiga kategori: (1) interpolasi

pada blok piksel di pojok, (2) interpolasi pada blok piksel di sisi, dan (3) interpolasi pada blok piksel yang berada di tengah. Berikut akan dijelaskan masing-masing dari teknik tersebut.

1. Teknik Interpolasi pada Blok Piksel di Pojok (*Corner Edge*)

Teknik ini dilakukan pada blok-blok piksel yang terletak di pojok citra. Hal ini menyebabkan jumlah piksel referensi yang digunakan untuk interpolasi hanya sedikit (Gambar 3.4). Sehingga memerlukan proporsi pembobotan yang berbeda dengan blok-blok piksel yang berada di posisi lain.

1	2	3	x
2			
3			
y			

Gambar 3.4 Posisi blok piksel di pojok citra

Secara matematis, syarat yang harus dipenuhi oleh sebuah piksel untuk melakukan operasi ini pada Persamaan 3.5. Posisi piksel yang dilakukan operasi ini seperti pada Gambar 3.5, ditunjukkan oleh piksel b dan g . Persamaan yang digunakan untuk masing-masing piksel tersebut ditampilkan pada Persamaan 3.6 dan 3.7. Pemberian nilai bobot untuk rata-rata piksel yang bertetangga dalam satu blok adalah 0.7, sedangkan untuk rata-rata piksel yang tidak satu blok adalah 0.3. Nilai ini diberikan setelah dilakukan proses penghitungan manual dan menggunakan metode *trial and error*. Sedangkan untuk piksel yang berada di sisi sebelah dalam, h dan l menggunakan Teknik Interpolasi pada blok piksel yang ada di sisi.

a	b	c	d	e	f
g	x	h	i	x	j
k	l	m	n	o	p
q	r	s	t	u	v
...	x	x	...
...

Gambar 3.5 Komponen blok piksel yang berada di pojok citra

$$x_n = \{1, x\}, y_n = \{1, y\} \quad (3.5)$$

$$IP_b = \left\lfloor \left(0.7 \times \left(\frac{a+c}{2} \right) \right) + (0.3 \times d) \right\rfloor \quad (3.6)$$

$$IP_g = \left\lfloor \left(0.7 \times \left(\frac{a+k}{2} \right) \right) + (0.3 \times q) \right\rfloor \quad (3.7)$$

2. Teknik Interpolasi pada Blok Piksel di Sisi (*Side Edge*)

Teknik ini dilakukan pada blok-blok piksel yang terletak di sisi-sisi citra. Karena jumlah piksel referensi yang digunakan untuk interpolasi sedikit lebih banyak daripada blok piksel yang berada di pojok (Gambar 3.4). Sehingga proporsi pembobotan pun berbeda dengan blok-blok piksel yang berada di posisi pojok.

1	2	3	x
2			
3			
y			

Gambar 3.6 Posisi blok piksel di sisi citra

Secara matematis, syarat yang harus dipenuhi oleh sebuah piksel untuk melakukan operasi ini pada Persamaan 3.8. Posisi piksel yang dilakukan operasi ini seperti pada Gambar 3.7, ditunjukkan oleh piksel *e*. Persamaan yang digunakan untuk piksel tersebut ditampilkan pada Persamaan 3.9. Sedangkan untuk operasi pada piksel *i* ditampilkan pada Persamaan 3.10. Piksel *h* dan *j* memiliki operasi yang serupa dengan piksel *i*.

a	b	c	d	e	f	q	r	s
g	x	h	i	x	j	...	x	...
k	l	m	n	o	p
...	...	t	u	v	w
...	x	x	x	...
...

Gambar 3.7 Komponen blok piksel yang berada di sisi citra

$$\begin{cases} x_n = \{2, \dots, x-1\}, y_n = \{1, y\} \\ x_n = \{1, x\}, y_n = \{2, \dots, y-1\} \end{cases} \quad (3.8)$$

$$IP_e = \left\lfloor \left(0.5 \times \left(\frac{d+f}{2} \right) \right) + (0.25 \times c) + (0.25 \times q) \right\rfloor \quad (3.9)$$

$$IP_i = \left\lfloor \left(0.5 \times \left(\frac{d+n}{2} \right) \right) + \left(0.3 \times \left(\frac{c+m}{2} \right) \right) + \left(0.2 \times \left(\frac{t+u}{2} \right) \right) \right\rfloor \quad (3.10)$$

Untuk piksel yang berada di tepi, bobot yang diberikan untuk piksel yang bertetangga satu blok adalah 0.5, sedangkan untuk yang tidak satu blok adalah 0.25. Namun piksel yang terletak di tengah seperti piksel h , i , dan j diberikan nilai bobot 0.5 untuk rata-rata piksel satu blok, 0.3 untuk piksel yang bertetangga namun tidak satu blok, dan 0.2 untuk rata-rata piksel yang bertetangga namun tidak bersinggungan langsung. Nilai ini diberikan setelah dilakukan proses penghitungan manual dan menggunakan metode *trial and error*.

3. Teknik Interpolasi pada Blok Piksel di Tengah

Teknik ini dilakukan pada blok-blok piksel yang terletak di tengah-tengah citra. Karena jumlah piksel referensi yang digunakan untuk interpolasi lebih banyak daripada blok piksel yang berada di sisi dan pojok (Gambar 3.8). Sehingga proporsi pembobotan pun berbeda dengan blok-blok piksel yang berada di posisi-posisi lain.

1	2	3	x
2			
3			
y			

Gambar 3.8 Posisi blok piksel di tengah-tengah

Secara matematis, syarat yang harus dipenuhi oleh sebuah piksel untuk melakukan operasi ini pada Persamaan 3.11. Posisi piksel yang dilakukan operasi ini seperti pada Gambar 3.9, ditunjukkan oleh piksel o . Persamaan yang digunakan untuk piksel tersebut ditampilkan pada Persamaan 3.12.

a	b	c	d	e	f
g	x	h	...	x	x	...
k	l	m	n	o	p	t	u	v
...	...	i	q	r	s	j
...	x	x	x	...
...

Gambar 3.9 Komponen blok piksel yang berada di tengah citra

$$x_n = \{2, \dots, x-1\}, y_n = \{2, \dots, y-1\} \quad (3.11)$$

$$IP_o = \left\lfloor \left(0.5 \times \left(\frac{n+p}{2} \right) \right) + \left(0.3 \times \left(\frac{q+s}{2} \right) \right) + \left(0.2 \times \left(\frac{m+i+t+j}{4} \right) \right) \right\rfloor \quad (3.12)$$

Bobot untuk rata-rata piksel yang bertetangga satu blok adalah 0.5, sedangkan untuk rata-rata piksel yang tidak satu blok namun bersisian adalah 0.3, sedangkan untuk rata-rata piksel yang tidak bersentuhan langsung adalah 0.2. Nilai ini diberikan setelah dilakukan proses penghitungan manual dan menggunakan metode *trial and error*.

3.3.2. Proses Penyisipan

Dalam metode yang diajukan, proses penyisipan dilakukan sebanyak dua kali. Pertama, penyisipan dilakukan pada piksel hasil interpolasi (IP). Setelah semua IP berhasil disisipkan pesan, maka proses penyisipan berpindah pada piksel referensi dari citra orisinil (RP).

1. Mencari selisih antara piksel hasil interpolasi dengan nilai piksel asli seperti pada Persamaan 3.13.

$$v_n = EP_n - IP_n \quad \text{dengan } n = 1, 2, 3, 4 \quad (3.13)$$

dengan:

v_n Selisih antara piksel hasil interpolasi dengan piksel asli pada piksel ke n .

EP_n Nilai piksel asli dari piksel ke n pada blok piksel tersebut.

IP_n Nilai piksel hasil interpolasi dari piksel ke n pada blok piksel tersebut.

RP_1	U_1	RP_2
U_2	U_m	U_3
RP_3	U_4	RP_4

Gambar 3.10 Pemetaan blok piksel sebelum proses penyisipan

2. Melakukan reduksi pada nilai selisih menggunakan Persamaan 3.14 (Ahmad, 2014).

$$\widetilde{v}_n = \begin{cases} v_n & \text{jika } -2 < v_n < 2 \\ \lfloor v_n + 2^{(\log_2 |v_n|) - 1} \rfloor & \text{jika } -2 \geq v_n \\ \lfloor v_n - 2^{(\log_2 |v_n|) - 1} \rfloor & \text{jika } 2 \leq v_n \end{cases} \quad (3.14)$$

dengan:

- v_n Selisih antara piksel hasil interpolasi dengan piksel asli pada piksel ke n .
- \widetilde{v}_n Selisih baru setelah memenuhi salah satu persyaratan di atas.

3. Mengubah pesan menjadi bilangan basis 2, kemudian melakukan ekspansi dan penyisipan beban pesan menggunakan Persamaan 3.15 pada selisih yang telah direduksi menggunakan Persamaan 3.14. Sehingga nilai piksel baru akan dipetakan seperti pada Gambar 3.10.

$$\widetilde{U}_n = 2 \times \widetilde{v}_n + b, \quad b \in \{0, 1\} \quad (3.15)$$

dengan:

- \widetilde{U}_n Nilai selisih baru yang telah dilakukan proses penyisipan pesan.
- \widetilde{v}_n Selisih baru setelah dilakukan persamaan reduksi selisih.
- b Nilai pesan rahasia.

4. Sehingga nilai piksel baru dari metode yang diusulkan ini seperti pada Persamaan 3.16.

$$\begin{cases} U'_m = U_m \\ U'_n = U_m + \widetilde{U}_n \end{cases} \quad (3.16)$$

dengan:

- U_m Nilai *base point* dari sebuah blok piksel (posisi yang berada di tengah-tengah).
- U'_m Nilai piksel baru dari *base point*.
- \widetilde{U}_n Nilai selisih baru yang telah dilakukan proses penyisipan pesan.
- U'_n Nilai piksel baru.

5. Pada tahap kedua, menghitung selisih antara keempat piksel referensi (RP) dengan piksel hasil interpolasi yang berada di sebelah kiri atau kanan piksel tersebut (U'_1 dan U'_4) menggunakan Persamaan 3.17 dengan sedikit perubahan dari penelitian milik (Govind, 2015).

RP_1	U'_1	RP_2
U'_2	U_m	U'_3
RP_3	U'_4	RP_4

Gambar 3.11 Gambar piksel yang digunakan untuk menentukan nilai prediksi

$$\begin{cases} dk_{1,2} = RP_{1,2} - U'_1 \\ dk_{3,4} = RP_{3,4} - U'_4 \end{cases} \quad (3.17)$$

dengan:

dk_n Nilai selisih sebuah piksel referensi n dengan piksel hasil interpolasi dalam satu blok.

$U'_{1,4}$ Nilai piksel hasil interpolasi yang telah disisipkan pesan rahasia pada indeks ke-1 dan ke-4.

6. Proses penyisipan adaptif dilakukan pada RP dengan melakukan *thresholding* pada selisih piksel yang didapat, apabila memenuhi Persamaan 3.18, akan dilakukan penyisipan menggunakan Persamaan 3.19. Sehingga, pemetaan hasil penyisipan pada kedua jenis piksel (IP dan RP) seperti pada Gambar 3.12.

$$|dk_n| \leq T \quad (3.18)$$

dengan:

dk_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

T Nilai batas untuk selisih piksel yang diperbolehkan untuk disisipi sebuah pesan.

$$RP'_n = U'_n + (2 \times dk_n + b), \quad b \in \{0, 1\} \quad (3.19)$$

dengan:

RP'_n Nilai piksel referensi baru setelah proses penyisipan.

U'_n Nilai piksel hasil interpolasi yang telah disisipkan pesan rahasia dan berada dalam satu blok.

dk_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

b Nilai pesan rahasia.

7. Apabila selisih nilai piksel dk tidak memenuhi Persamaan 3.18 seperti pada Persamaan 3.20. Maka akan dilakukan penggeseran satu bit seperti pada Persamaan 3.21 berikut.

$$|dk_n| > T \quad (3.20)$$

dengan:

dk_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

T Nilai batas untuk selisih piksel yang diperbolehkan untuk disisipi sebuah pesan.

$$RP'_n = \begin{cases} RP_n - (T + 1) & \text{jika } dk_n \geq 0 \\ RP_n + (T + 1) & \text{jika } dk_n < 0 \end{cases} \quad (3.21)$$

dengan:

RP'_n Nilai piksel referensi baru setelah proses penyisipan.

dk_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

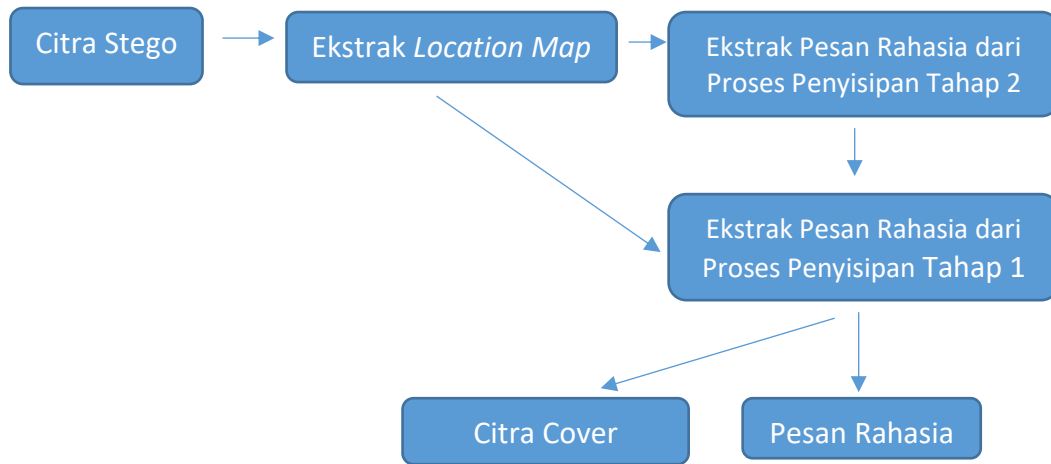
T Nilai batas untuk selisih piksel yang diperbolehkan untuk disisipi sebuah pesan.

RP'_1	U'_1	RP'_2
U'_2	U_m	U'_3
RP'_3	U'_4	RP'_4

Gambar 3.12 Pemetaan hasil akhir penyisipan pada blok piksel

3.3.3. Proses Ekstraksi dan Pengembalian Citra *Cover*

Proses ekstraksi dilakukan secara sekuensial, diawali dengan mengekstrak pesan rahasia yang disimpan pada piksel referensi. Alur proses dari tahap ekstraksi dapat dilihat pada Gambar 3.13.



Gambar 3.13 Diagram Alir Proses Ekstraksi

Berikut proses ekstraksi pada metode steganografi yang diusulkan:

1. Menghitung selisih antara piksel referensi dengan piksel hasil interpolasi dalam setiap blok seperti pada Persamaan 3.22.

$$dk'_n = RP'_n - U'_n \quad (3.22)$$

dengan:

dk'_n Nilai selisih sebuah piksel referensi n yang dengan piksel hasil interpolasi dalam satu blok.

RP'_n Nilai piksel referensi.

U'_n Nilai piksel hasil interpolasi yang telah disisipkan pesan rahasia.

2. Melakukan proses ekstraksi tahap dua pada piksel referensi. Apabila nilai absolut selisih piksel memenuhi Persamaan 3.23, maka akan dalam piksel tersebut menyimpan piksel rahasia.

$$|dk'_n| \leq 2 \times T + 1 \quad (3.23)$$

dengan:

dk'_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

T Nilai batas untuk selisih piksel yang diperbolehkan untuk disisipi sebuah pesan.

3. Mengambil pesan rahasia dari piksel RP'_n yang memenuhi Persamaan 3.23 dengan menggunakan Persamaan 3.24.

$$b_n = (RP'_n - U'_n) \bmod 2, \quad b \in \{0, 1\} \quad (3.24)$$

dengan:

RP'_n Nilai piksel referensi.

U'_n Nilai piksel hasil interpolasi yang telah disisipkan pesan rahasia.

b Nilai pesan rahasia.

4. Setelah berhasil mengambil pesan rahasia pada piksel referensi, dilakukan proses pengembalian nilai piksel tersebut menggunakan Persamaan 3.25.

$$RP_n = \begin{cases} U'_n + \left\lfloor \frac{|dk'_n|}{2} \right\rfloor & \text{jika } RP'_n \geq U'_n \\ U'_n - \left\lfloor \frac{|dk'_n|}{2} \right\rfloor & \text{jika } RP'_n < U'_n \end{cases} \quad (3.25)$$

dengan:

RP_n Nilai piksel referensi asli.

U'_n Nilai piksel hasil interpolasi yang telah disisipkan pesan rahasia.

dk'_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

5. Apabila nilai selisih piksel tidak memenuhi Persamaan 3.23, namun memenuhi Persamaan 3.26. Maka piksel tersebut tidak menyimpan pesan rahasia. Sehingga nilai piksel referensi aslinya dapat dikembalikan menggunakan Persamaan 3.27.

$$|dk'_n| > 2 \times T + 1 \quad (3.26)$$

dengan:

dk'_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

T Nilai batas untuk selisih piksel yang diperbolehkan untuk disisipi sebuah pesan.

$$RP_n = \begin{cases} RP'_n - (T + 1) & \text{jika } RP'_n \geq U'_n \\ RP'_n + (T + 1) & \text{jika } RP'_n < U'_n \end{cases} \quad (3.27)$$

dengan:

RP'_n Nilai piksel referensi baru setelah proses penyisipan.

dk_n Nilai selisih antara nilai piksel referensi dengan nilai piksel hasil interpolasi.

T Nilai batas untuk selisih piksel yang diperbolehkan untuk disisipi sebuah pesan.

6. Membaca file *location map* untuk mengetahui apakah jenis reduksi selisih pada piksel hasil interpolasi yang dilakukan. Mengambil nilai selisih yang menyimpan pesan rahasia dengan Persamaan 3.28.

$$\widetilde{U}_n = U'_n - U_m \quad (3.28)$$

dengan:

- U_m Nilai *base point* dari sebuah blok piksel (posisi yang berada di tengah-tengah).
- \widetilde{U}_n Nilai selisih yang telah dilakukan proses penyisipan pesan.
- U'_n Nilai piksel yang masih menyimpan pesan rahasia.

7. Mengambil pesan dari piksel hasil interpolasi dengan menggunakan Persamaan 3.29. Kemudian mencari nilai selisih yang digunakan menggunakan Persamaan 3.30.

$$b = \widetilde{U}_n \bmod 2, b \in \{0, 1\} \quad (3.29)$$

$$\widetilde{v}_n = \frac{\widetilde{U}_n - b}{2} \quad (3.30)$$

dengan:

- b Nilai pesan rahasia.
- \widetilde{U}_n Nilai selisih yang telah dilakukan proses penyisipan pesan.
- \widetilde{v}_n Nilai selisih yang telah direduksi dan telah dilakukan proses ekstraksi.

8. Pada tahap kedua ini, *location map* akan dibaca untuk mengembalikan nilai selisih piksel yang asli atau sebelum dilakukan reduksi. Menggunakan Tabel 3.1, nilai selisih asli piksel akan dikembalikan pada Persamaan 3.31 hingga 3.33.

$$v_n = \widetilde{v}_n \text{ jika } LM = 001 \quad (3.31)$$

$$v_n = \begin{cases} \widetilde{v}_n + 2^{\log_2 |\widetilde{v}_n|} & \text{jika } LM = 010 \\ \widetilde{v}_n + 2^{(\log_2 |\widetilde{v}_n|) + 1} & \text{jika } LM = 011 \end{cases} \quad (3.32)$$

$$v_n = \begin{cases} \widetilde{v}_n - 2^{\log_2 |\widetilde{v}_n|} & \text{jika } LM = 100 \\ \widetilde{v}_n - 2^{(\log_2 |\widetilde{v}_n|) - 1} & \text{jika } LM = 101 \end{cases} \quad (3.33)$$

9. Setelah berhasil mengembalikan nilai selisih asli piksel hasil interpolasi, maka untuk mengembalikan citra asli dilakukan proses interpolasi menggunakan nilai-nilai dari piksel referensi yang telah dikembalikan. Seperti pada Persamaan 3.5 hingga 3.12.
10. Apabila semua nilai hasil interpolasi diketahui, maka nilai piksel asli dikembalikan dengan menggunakan Persamaan 3.34.

$$EP_n = IP_n + v_n \quad (3.34)$$

3.3.4. *Location Map*

Dalam metode yang diusulkan ini, *location map* dibutuhkan untuk menyimpan informasi mengenai jenis reduksi pada piksel hasil interpolasi. Pada penyisipan tahap pertama, diperlukan *location map* untuk memberi tanda piksel mana yang direduksi dan mana yang tidak. Pada tahap ini terdapat dua bit *location map* yang digunakan untuk menandai jenis reduksi selisih yang digunakan oleh setiap piksel dan dijelaskan pada Tabel 3.1.

Pada jenis reduksi ‘tetap’, digunakan untuk menandai piksel-piksel yang selisihnya tidak direduksi. Sedangkan untuk jenis reduksi ‘Positif dan Ganjil’ merupakan penanda untuk piksel yang selisih aslinya bernilai positif dan ganjil yang telah dilakukan reduksi. Begitu pula dengan jenis reduksi ‘Positif dan Genap’.

Sedangkan untuk jenis reduksi ‘Negatif dan Ganjil’ digunakan untuk menandai selisih piksel yang bernilai negatif dan ganjil. Untuk jenis reduksi ‘Negatif dan Genap’ memiliki perbedaan pada nilainya yang genap.

Tabel 3.1 Model *Location Map*

Jenis Reduksi	Bit Pertama	Bit Kedua	Bit Ketiga
Tetap	0	0	1
Positif dan Gasal	0	1	0
Positif dan Genap	0	1	1
Negatif dan Gasal	1	0	0
Negatif dan Genap	1	0	1

Pada Tabel 3.1 bit yang digunakan untuk menandai bahwa sebuah selisih tidak direduksi dan bernilai tetap dengan bit pertama dan kedua bernilai nol, sedangkan bit ketiganya memiliki nilai satu. Sedangkan untuk selisih yang direduksi namun bernilai positif, maka bit pertama bernilai nol dan bit keduanya bernilai satu. Akan tetapi perbedaan pada bit ketiga ditunjukkan apabila selisih bernilai gasal, maka bit ketiganya bernilai nol, namun apabila bernilai positif maka bit ketiganya bernilai satu.

Nilai selisih yang direduksi namun bernilai negatif akan ditandai dengan bit pertama bernilai satu dan bit kedua bernilai nol. Sedangkan untuk bit ketiganya akan bernilai nol jika gasal, dan bernilai satu apabila genap.

3.4 Implementasi Penelitian

Lingkungan implementasi dilakukan pada Matlab versi 2015a dengan spesifikasi komputer Prosesor AMD A8-7410 dan RAM 4GB. Pesan rahasia yang akan disisipkan adalah teks standar untuk penerbitan yaitu "*Lorem Ipsum*". Citra digital yang digunakan adalah citra *testing* standar berwarna *greyscale* berukuran 512×512 piksel, antara lain: *Lena*, *Baboon*, *Plane*, *Boat*, dan *Peppers* (Carolina, 2015).

3.5 Perancangan Uji Coba dan Analisis Hasil

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang akan dilakukan pada penelitian ini. Skenario uji coba yang akan dilakukan untuk menguji metode ini adalah sebagai berikut:

1. Skenario 1

Seluruh citra *cover* disisipi dengan 25,000 bit pesan setiap langkah hingga kapasitas maksimal. Kemudian dilakukan pengukuran kualitas menggunakan PSNR, SSIM, UQI. Setelah mendapatkan semua parameter pengukuran kualitas citra, maka dibandingkan dengan nilai PSNR, SSIM, dan UQI dari citra *cover* sebelum dilakukan penyisipan.

2. Skenario 2

Seluruh citra *cover* disisipi dengan 50,000 bit, 100,000 bit, hingga jumlah maksimum kapasitas citra yang dapat disisipi. Selanjutnya dilakukan steganalisis pada citra stego tersebut dengan melakukan pengukuran *K-L Divergence*.

3. Skenario 3

Dilakukan penggantian nilai pembobotan dalam proses interpolasi untuk mengetahui nilai yang mendapatkan hasil paling maksimal. Pengujian dilakukan dengan mencari nilai PSNR, SSIM, dan UQI dalam setiap iterasi penggantian nilai menggunakan metode *trial and error*.

4. Skenario 4

Dilakukan penggantian nilai *thresholding* dalam proses penyisipan tahap dua untuk mengetahui kapasitas paling maksimal. Pengujian dilakukan dengan mencari nilai PSNR, SSIM, UQI, dan jumlah BPP dalam setiap iterasi penggantian nilai.

PSNR digunakan untuk menyelidiki kemiripan antara dua citra dan seberapa mirip kedua citra tersebut, antara citra *cover* dengan citra stego yang telah mengalami proses interpolasi dan penyisipan pesan. Sedangkan SSIM digunakan untuk memeriksa kemiripan secara struktural dari dua buah citra yang dibandingkan. Perbedaan terdapat pada aspek-aspek yang dibandingkan. SSIM menggunakan iluminasi, kontras, dan citra (Wang, 2004). Namun PSNR menggunakan perbandingan terbalik dari MSE yang menghitung perbedaan dari kedua citra (Almohammad, 2010).

UQI merupakan sebuah indeks yang dapat memeriksa tingkat kemiripan dengan cara yang paling sederhana. Indeks ini tidak bergantung pada kondisi saat melihat citra. Secara matematis indeks ini akan menghitung: terdapatnya korelasi, distorsi pencahayaan, dan distorsi kontras. Ketiga parameter pengukuran ini digunakan untuk memeriksa apakah terdapat perbedaan yang jauh antara citra *cover* dengan citra stego yang telah mengalami proses interpolasi dan penyisipan pesan.

Pada bagian analisa hasil akan dibandingkan parameter pengukuran kualitas citra tersebut, sehingga dapat diketahui apakah metode yang diusulkan lebih baik dari sisi kapasitas dan kualitas dengan metode sebelumnya. Apabila nilai

dari parameter-parameter yang ditentukan meningkat, maka metode yang diusulkan telah berhasil memenuhi tujuan.

3.6 Penyusunan Buku Tesis

Tahap ini merupakan proses pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, dan proses yang telah dilakukan. Tidak lupa mencantumkan hasil-hasil dan analisis yang telah didapatkan selama penelitian. Buku tesis yang akan disusun bertujuan untuk memberikan gambaran dari pengerjaan tesis dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan penelitian lebih lanjut.

3.7 Jadwal Penelitian

Pada bagian ini akan ditampilkan jadwal kegiatan selama penelitian. Jadwal yang disajikan selama tiga bulan, dimulai dari bulan Februari tahun 2016 hingga April tahun 2016. Penjelasan jadwal secara rinci mengenai jadwal kegiatan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Jadwal Pelaksanaan Penelitian

Aktivitas	Bulan 1				Bulan 2				Bulan 3			
Studi Literatur												
Perancangan Algoritma												
Implementasi Algoritma												
Pengujian												
Analisis Hasil												
Penyusunan Buku Tesis												

[Halaman ini sengaja dikosongkan]

BAB 4

PEMBAHASAN DAN ANALISA HASIL UJI COBA

Pada bab ini akan dijelaskan bagaimana penelitian ini dilakukan sesuai dengan metode yang telah diajukan sebelumnya. Sedangkan pada subbab selanjutnya, akan dijelaskan hasil uji coba beserta analisisnya.

4.1. Pembahasan

Dalam subbab ini akan dijelaskan penulisan kode penerapan dari metode yang telah diajukan. Penjelasan akan dibagi menjadi tiga bagian, yaitu proses interpolasi, proses penyisipan tahap pertama, dan proses penyisipan tahap kedua.

4.1.1. Proses Interpolasi

Pada proses ini dilakukan konstruksi nilai piksel-piksel baru yang nantinya digunakan untuk Proses Penyisipan Tahap Pertama. Seperti pada bab sebelumnya yang menjelaskan bagaimana proses interpolasi dilakukan. Proses ini terbagi menjadi tiga tahap lagi, yaitu Proses Interpolasi pada *Piksel* Pojok, Proses Interpolasi pada *Piksel* Sisi, dan Proses Interpolasi pada *Piksel* Tengah.

Pada Gambar 4.1 ditampilkan kode untuk implementasi proses interpolasi pada piksel sebuah citra yang berada di pojok kiri atas. Pada baris ke-11 hingga ke-14 merupakan penerapan dari Persamaan 3.5 dan 3.7. Kemudian dilakukan penggantian nilai piksel pada baris ke-15 hingga ke-18.

Gambar 4.1 Kode untuk Proses Interpolasi Pada Piksel Pojok Kiri Atas

```
1  if x==1 && y==1 %edge pojok kiri atas
2  d1=double(d);
3  f1=double(f);
4  j1=double(j);
5  l1=double(l);
6  f2=double(A(b,c+3));
7  j2=double(A(b+3,c));
8  l2=double(A(b+2,c+3));
9  n1=double(A(b+3,c+2));
10 n2=double(A(b+3,c+3));
11 edge1=fix(((0.7*((d1+f1)/2))+(0.3*f2));
```

```

12 edge2=fix(((0.7*((d1+j1))/2)+(0.3*j2));
13 nedge1=fix(((0.5*((f1+l1))/2)+(0.3*((f2+l2))/2)+(0.2*((n1+
n2)/2)));
14 nedge2=fix((0.5*((j1+l1)/2))+
(0.3*((j2+n1)/2)+(0.2*((l2+n2)/2)));
15 Z(b,c+1)=uint8(edge1);
16 Z(b+1,c)=uint8(edge2);
17 Z(b+1,c+2)=uint8(nedge1);
18 Z(b+2,c+1)=uint8(nedge2);
19 end

```

Sedangkan pada Gambar 4.2 ditampilkan kode untuk implementasi proses interpolasi pada piksel sebuah citra yang berada di pojok kanan atas. Pada baris ke-11 hingga ke-14 merupakan penerapan dari Persamaan 3.5 dan 3.7. Kemudian dilakukan penggantian nilai piksel pada baris ke-15 hingga ke-18.

Gambar 4.2 Kode untuk Proses Interpolasi Pada Piksel Pojok Kanan Atas

```

1  if x==170 && y==1 %edge pojok kanan atas
2  d1=double(d);
3  f1=double(f);
4  l1=double(l);
5  j1=double(j);
6  d2=double(A(b,c-1));
7  j2=double(A(b+2,c-1));
8  l2=double(A(b+3,c+2));
9  n1=double(A(b+3,c-1));
10 n2=double(A(b+3,c));
11 edge3=fix(((0.7*((d1+f1))/2)+(0.3*d2));
12 edge4=fix(((0.7*((f1+l1))/2)+(0.3*l2));
13 nedge3=fix((0.5*((d1+j1)/2)+(0.3*((d2+j2)/2)+(0.2*((n1+n2)
)/2)));
14 nedge4=fix((0.5*((j1+l1)/2)+(0.3*((l2+n2)/2)+(0.2*((j2+n1)
)/2)));
15 Z(b,c+1)=uint8(edge3);
16 Z(b+1,c)=uint8(nedge3);
17 Z(b+1,c+2)=uint8(edge4);
18 Z(b+2,c+1)=uint8(nedge4);
19 end

```

Implementasi kode proses interpolasi yang berada di pojok kanan bawah ditampilkan pada Gambar 4.3. Pada baris ke-11 hingga ke-14 merupakan penerapan dari Persamaan 3.5 dan 3.7. Kemudian dilakukan penggantian nilai piksel pada baris ke-15 hingga ke-18.

Gambar 4.3 Kode untuk Proses Interpolasi Pada Pixel Pojok Kanan Bawah

```
1  if x==170 && y==170 %edge pojok kanan bawah
2  d1=double(d);
3  f1=double(f);
4  j1=double(j);
5  l1=double(l);
6  d2=double(A(b-1,c));
7  f2=double(A(b-1,c+2));
8  j2=double(A(b+2,c-1));
9  n1=double(A(b-1,c-1));
10 n2=double(A(b,c-1));
11 edge7=fix((0.7*((f1+l1)/2))+(0.3*f2));
12 edge8=fix((0.7*((j1+l1)/2))+(0.3*j2));
13 nedge7=fix((0.5*((d1+f1)/2))+(0.3*((d2+f2)/2))+(0.2*((n1+n2)/2));
14 nedge8=fix((0.5*((d1+j1)/2))+(0.3*((n2+j2)/2))+(0.2*((n1+d2)/2));
15 Z(b,c+1)=uint8(nedge7);
16 Z(b+1,c)=uint8(nedge8);
17 Z(b+1,c+2)=uint8(edge7);
18 Z(b+2,c+1)=uint8(edge8);
19 end
```

Implementasi kode proses interpolasi yang berada di pojok kiri bawah ditampilkan pada Gambar 4.4. Pada baris ke-11 hingga ke-14 merupakan penerapan dari Persamaan 3.5 dan 3.7. Kemudian dilakukan penggantian nilai piksel pada baris ke-15 hingga ke-18.

Gambar 4.4 Kode untuk Proses Interpolasi Pada Pixel Pojok Kiri Bawah

```
1  if x==1 && y==170 %edge pojok kiri bawah
2  d1=double(d);
3  f1=double(f);
4  j1=double(j);
5  l1=double(l);
6  d2=double(A(b-1,c));
7  l2=double(A(b+2,c+3));
8  f2=double(A(b,c+3));
9  n1=double(A(b-1,c+2));
10 n2=double(A(b-1,c+3));
11 edge5=fix((0.7*((d1+j1)/2))+(0.3*d2));
12 edge6=fix((0.7*((j1+l1)/2))+(0.3*l2));
13 nedge5=fix((0.5*((d1+f1)/2))+(0.3*((d2+n1)/2))+(0.2*((n2+f2)/2));
14 nedge6=fix((0.5*((f1+l1)/2))+(0.3*((f2+l2)/2))+(0.2*((n1+n2)/2));
15 Z(b,c+1)=uint8(nedge5);
16 Z(b+1,c)=uint8(edge5);
17 Z(b+1,c+2)=uint8(nedge6);
18 Z(b+2,c+1)=uint8(edge6);
19 end
```

Pada Gambar 4.5 ditampilkan kode untuk implementasi proses interpolasi pada piksel sebuah citra yang berada di sisi atas. Pada baris ke-15 hingga ke-18 merupakan penerapan dari Persamaan 3.9 dan 3.10. Kemudian dilakukan penggantian nilai piksel pada baris ke-19 hingga ke-22.

Gambar 4.5 Kode untuk Proses Interpolasi Pada Piksel Sisi Atas

```

1  if y==1 %sisi atas
2  if x>=2 && x<=169
3  d1=double(d);
4  f1=double(f);
5  j1=double(j);
6  l1=double(l);
7  d2=double(A(b,c-1));
8  f2=double(A(b,c+3));
9  j2=double(A(b+2,c-1));
10 l2=double(A(b+2,c+3));
11 n1=double(A(b+3,c-1));
12 n2=double(A(b+3,c));
13 n3=double(A(b+3,c+2));
14 n4=double(A(b+3,c+3));
15 edge9=fix((0.5*((d1+f1)/2))+(0.25*d2)+(0.25*f2));
16 nedge9=fix((0.5*((d1+j1)/2))+(0.3*((d2+j2)/2))+(0.2*((n1+n2)/2));
17 nedge10=fix((0.5*((f1+l1)/2))+(0.3*((f2+l2)/2))+(0.2*((n3+n4)/2));
18 nedge11=fix((0.5*((j1+l1)/2))+(0.3*((n2+n3)/2))+(0.2*((j2+l2+n1+n4)/4));
19 Z(b,c+1)=uint8(edge9);
20 Z(b+1,c)=uint8(nedge9);
21 Z(b+1,c+2)=uint8(nedge10);
22 Z(b+2,c+1)=uint8(nedge11);
23 end
24 end

```

Sedangkan pada Gambar 4.6 ditampilkan kode untuk implementasi proses interpolasi pada piksel sebuah citra yang berada di sisi kiri. Pada baris ke-15 hingga ke-18 merupakan penerapan dari Persamaan 3.9 dan 3.10. Kemudian dilakukan penggantian nilai piksel pada baris ke-19 hingga ke-22.

Gambar 4.6 Kode untuk Proses Interpolasi Pada Piksel Sisi Kiri

```

1  if x==1 %sisi kiri
2  if y>=2 && y<=169
3  d1=double(d);
4  f1=double(f);
5  j1=double(j);
6  l1=double(l);

```



```

7   d2=double(A(b-1,c));
8   f2=double(A(b-1,c+2));
9   j2=double(A(b+3,c));
10  l2=double(A(b+3,c+2));
11  n1=double(A(b-1,c+3));
12  n2=double(A(b,c+3));
13  n3=double(A(b+2,c+3));
14  n4=double(A(b+3,c+3));
15  edge10=fix((0.5*((d1+j1)/2))+(0.25*d2)+(0.25*j2));
16  nedge12=fix((0.5*((d1+f1)/2))+(0.3*((d2+f2)/2))+(0.2*((n1+n
    2)/2)));
17  nedge13=fix((0.5*((f1+l1)/2))+(0.3*((n2+n3)/2))+(0.2*((f2+n
    1+l2+n4)/4)));
18  nedge14=fix((0.5*((j1+l1)/2))+(0.3*((j2+l2)/2))+(0.2*((n3+n
    4)/2)));
19  Z(b,c+1)=uint8(nedge12);
20  Z(b+1,c)=uint8(edge10);
21  Z(b+1,c+2)=uint8(nedge13);
22  Z(b+2,c+1)=uint8(nedge14);
23  end
24  end

```

Implementasi kode proses interpolasi blok piksel yang berada di sisi kanan citra ditampilkan pada Gambar 4.7. Pada baris ke-15 hingga ke-18 merupakan penerapan dari Persamaan 3.9 dan 3.10. Kemudian dilakukan penggantian nilai piksel pada baris ke-19 hingga ke-22.

Gambar 4.7 Kode untuk Proses Interpolasi Pada Piksel Sisi Kanan

```

1  if x==170 %sisi kanan
2  if y>=2 && y<=169
3  d1=double(d);
4  f1=double(f);
5  j1=double(j);
6  l1=double(l);
7  d2=double(A(b-1,c));
8  f2=double(A(b-1,c+2));
9  j2=double(A(b+3,c));
10 l2=double(A(b+3,c+2));
11 n1=double(A(b-1,c-1));
12 n2=double(A(b,c-1));
13 n3=double(A(b+2,c-1));
14 n4=double(A(b+3,c-1));
15 edge11=fix((0.5*((f1+l1)/2))+(0.25*f2)+(0.25*l2));
16 nedge15=fix((0.5*((d1+f1)/2))+(0.3*((d2+f2)/2))+(0.2*((n1+n2)
    /2)));
17 nedge16=fix((0.5*((d1+j1)/2))+(0.3*((n2+n3)/2))+(0.2*((n1+d2+
    n4+j2)/4)));
18 nedge17=fix((0.5*((j1+l1)/2))+(0.3*((j2+l2)/2))+(0.2*((n3+n4)
    /2)));
19 Z(b,c+1)=uint8(nedge15);
20 Z(b+1,c)=uint8(nedge16);
21 Z(b+1,c+2)=uint8(edge11);

```

```

22 Z(b+2,c+1)=uint8(nedge17);
23 end
24 end

```

Kode proses interpolasi blok piksel yang berada di sisi bawah citra diimplementasikan seperti pada Gambar 4.8. Pada baris ke-15 hingga ke-18 merupakan penerapan dari Persamaan 3.9 dan 3.10. Kemudian dilakukan penggantian nilai piksel pada baris ke-19 hingga ke-22.

Gambar 4.8 Kode untuk Proses Interpolasi Pada Piksel Sisi Bawah

```

1  if y==170 %sisi bawah
2  if x>=2 && x<=169
3  d1=double(d);
4  f1=double(f);
5  j1=double(j);
6  l1=double(l);
7  d2=double(A(b,c-1));
8  f2=double(A(b,c+3));
9  j2=double(A(b+2,c-1));
10 l2=double(A(b+2,c+3));
11 n1=double(A(b-1,c-1));
12 n2=double(A(b-1,c));
13 n3=double(A(b-1,c+2));
14 n4=double(A(b-1,c+3));
15 edge12=fix((0.5*((j1+l1)/2))+(0.25*j2)+(0.25*l2));
16 nedge18=fix((0.5*((d1+f1)/2))+(0.3*((n2+n3)/2))+(0.2*((n1+d2+
    n4+f2)/4)));
17 nedge19=fix((0.5*((d1+j1)/2))+(0.3*((d2+j2)/2))+(0.2*((n1+n2)
    /2)));
18 nedge20=fix((0.5*((f1+l1)/2))+(0.3*((f2+l2)/2))+(0.2*((n3+n4)
    /2)));
19 Z(b,c+1)=uint8(nedge18);
20 Z(b+1,c)=uint8(nedge19);
21 Z(b+1,c+2)=uint8(nedge20);
22 Z(b+2,c+1)=uint8(edge12);
23 end
24 end

```

Sedangkan pada Gambar 4.9 ditampilkan kode untuk implementasi proses interpolasi pada piksel sebuah citra yang berada di tengah. Pada baris ke-19 hingga ke-22 merupakan penerapan dari Persamaan 3.12. Kemudian dilakukan penggantian nilai piksel pada baris ke-23 hingga ke-26.

Gambar 4.9 Kode untuk Proses Interpolasi Pada Piksel Tengah

```

1  if x>=2 && x<=169 %piksel tengah
2  if y>=2 && y<=169

```

```

3  d1=double(d);
4  f1=double(f);
5  j1=double(j);
6  l1=double(l);
7  d2=double(A(b-1,c));
8  d3=double(A(b,c-1));
9  f2=double(A(b-1,c+2));
10 f3=double(A(b,c+3));
11 j2=double(A(b+3,c));
12 j3=double(A(b+2,c-1));
13 l2=double(A(b+3,c+2));
14 l3=double(A(b+2,c+3));
15 n1=double(A(b-1,c-1));
16 n2=double(A(b-1,c+3));
17 n3=double(A(b+3,c-1));
18 n4=double(A(b+3,c+3));
19 nedge21=fix((0.5*((d1+f1)/2))+(0.3*((d2+f2)/2))+(0.2*((n1+n
2+d3+f3)/4)));
20 nedge22=fix((0.5*((d1+j1)/2))+(0.3*((d3+j3)/2))+(0.2*((n1+d
2+n3+j2)/4)));
21 nedge23=fix((0.5*((f1+l1)/2))+(0.3*((f3+l3)/2))+(0.2*((f2+n
2+l2+n4)/4)));
22 nedge24=fix((0.5*((j1+l1)/2))+(0.3*((j2+l2)/2))+(0.2*((j3+n
3+l3+n4)/4)));
23 Z(b,c+1)=uint8(nedge21);
24 Z(b+1,c)=uint8(nedge22);
25 Z(b+1,c+2)=uint8(nedge23);
26 Z(b+2,c+1)=uint8(nedge24);
27 end
28 end

```

4.1.2. Proses Penyisipan Tahap Pertama

Setelah semua blok piksel berhasil dilakukan proses interpolasi, langkah selanjutnya adalah mencari selisih antara piksel asli dan piksel hasil interpolasi citra. Kemudian selisih ini yang akan digunakan untuk penyisipan citra pada tahap pertama.

Gambar 4.10 Posisi Piksel dalam Sebuah Blok Piksel

RP_1	U_1	RP_2
U_2	U_m	U_3
RP_3	U_4	RP_4

Pada Gambar 4.11 merupakan kode implementasi dari proses pencarian selisih. Baris ke-21 merupakan selisih antara piksel asli dan piksel hasil interpolasi dari U_1 . Baris ke-30 adalah untuk piksel U_2 , baris ke-39 untuk piksel U_3 , dan baris

ke-48 untuk piksel U_4 . Posisi masing-masing piksel dalam sebuah blok dapat dilihat pada Gambar 4.10.

Gambar 4.11 Kode untuk Mencari Selisih antara Piksel Asli dan Piksel Hasil Interpolasi Citra

```
1  %mencari selisih interpolasi
2  d2=double (A (b,c) );
3  e2=double (A (b,c+1) );
4  f2=double (A (b,c+2) );
5  g2=double (A (b+1,c) );
6  h2=double (A (b+1,c+1) );
7  i2=double (A (b+1,c+2) );
8  j2=double (A (b+2,c) );
9  k2=double (A (b+2,c+1) );
10 l2=double (A (b+2,c+2) );
11 d3=double (Z (b,c) );
12 e3=double (Z (b,c+1) );
13 f3=double (Z (b,c+2) );
14 g3=double (Z (b+1,c) );
15 h3=double (Z (b+1,c+1) );
16 i3=double (Z (b+1,c+2) );
17 j3=double (Z (b+2,c) );
18 k3=double (Z (b+2,c+1) );
19 l3=double (Z (b+2,c+2) );
20
21 v1=fix (e2-e3) ;
22 if v1>-2 && v1<2
23 v11=v1;
24 elseif v1>=2
25 v11=fix (v1-2^ ( (log2 (abs (v1) ) ) -1) );
26 elseif v1<=-2
27 v11=fix (v1+2^ ( (log2 (abs (v1) ) ) -1) );
28 end
29
30 v2=fix (g2-g3) ;
31 if v2>-2 && v2<2
32 v21=v2;
33 elseif v2>=2
34 v21=fix (v2-2^ (log2 (abs (v2) ) -1) );
35 elseif v2<=-2
36 v21=fix (v2+2^ (log2 (abs (v2) ) -1) );
37 end
38
39 v3=fix (i2-i3) ;
40 if v3>-2 && v3<2
41 v31=v3;
42 elseif v3>=2
43 v31=fix (v3-2^ (log2 (abs (v3) ) -1) );
44 elseif v3<=-2
45 v31=fix (v3+2^ (log2 (abs (v3) ) -1) );
46 end
47
48 v4=fix (k2-k3) ;
```

```

49  if v4>-2 && v4<2
50  v41=v4;
51  elseif v4>=2
52  v41=fix(v4-2^(log2(abs(v4))-1));
53  elseif v4>=-2
54  v41=fix(v4+2^(log2(abs(v4))-1));
55  end

```

Setelah diketahui selisihnya, kemudian dilakukan reduksi pada masing-masing selisih yang terdapat pada baris ke-22 hingga ke-28 untuk \widetilde{v}_1 , baris ke-31 hingga ke-37 untuk \widetilde{v}_2 , baris ke-40 hingga ke-46 untuk \widetilde{v}_3 , dan baris ke-49 hingga ke-55 untuk \widetilde{v}_4 . Semua baris tersebut merupakan implementasi dari Persamaan 3.14.

Sedangkan Gambar 4.12 merupakan Kode untuk penyisipan tahap pertama setelah semua piksel asli dan hasil interpolasi telah diketahui selisihnya dan telah direduksi. Pada baris ketiga hingga ke-6 merupakan penyisipan yang dilakukan pada piksel U_1 hingga U_4 menggunakan nilai selisih baru yang didapatkan dari Gambar 4.11. Merupakan implementasi dari Persamaan 3.15.

Gambar 4.12 Kode untuk Penyisipan Tahap Pertama

```

1  for il=1:length(msga)
2  if il<=115600
3  if blokk==ceil(il/4) && mod(il,4)==1
    ZZ(b,c+1)=uint8(hhh+(2*(cz(ceil(il/4),5,1)))+(msga(1,il))));
4  elseif blokk==ceil(il/4) && mod(il,4)==2
    ZZ(b+1,c)=uint8(hhh+(2*(cz(ceil(il/4),7,1)))+(msga(1,il))));
    elseif ceil(il/4)==blokk && mod(il,4)==3
5  ZZ(b+1,c+2)=uint8(hhh+(2*(cz(ceil(il/4),9,1)))+(msga(1,il))
    );
6  elseif ceil(il/4)==blokk && mod(il,4)==0
    ZZ(b+2,c+1)=uint8(hhh+(2*(cz(ceil(il/4),11,1)))+(msga(1,il))
    ));
7  end
8  end
9  end

```

4.1.3. Proses Penyisipan Tahap Kedua

Apabila sebuah pesan memiliki kapasitas yang melebihi kapasitas pesan pada tahap pertama yaitu 115,600 bit. Maka akan dilakukan penyisipan tahap kedua. Dalam tahap ini menggunakan kode pada Gambar 4.11 untuk memeriksa

selisih antara piksel referensi dengan hasil interpolasi. Kemudian menggunakan kode pada Gambar 4.12 untuk proses penyisipan tahap kedua.

Gambar 4.11 Kode untuk Mencari Selisih antara Piksel Referensi dengan Piksel Hasil Interpolasi

```

1  for il=1:115600
2  if blk==ceil(il/4) && mod(il,4)==1
    dk1=(double(ZZ(b,c)))-(double(ZZ(b,c+1)));
3  disp (il);
4  cd(il,1)=il;
5  cd(il,2)=dk1;
6  cd(il,3)=ZZ(b,c);
7  cd(il,4)=ZZ(b,c+1);

8  elseif blk==ceil(il/4) && mod(il,4)==2
9  dk2=(double(ZZ(b,c+2)))-(double(ZZ(b,c+1)));
10 cd(il,1)=il;
11 cd(il,2)=dk2;
12 cd(il,3)=ZZ(b,c+2);
13 cd(il,4)=ZZ(b,c+3);

14 elseif ceil(il/4)==blk && mod(il,4)==3
15 dk3=(double(ZZ(b+2,c)))-(double(ZZ(b+2,c+1)));
16 cd(il,1)=il;
17 cd(il,2)=dk3;
18 cd(il,3)=ZZ(b+2,c);
19 cd(il,4)=ZZ(b+2,c+1);

20 elseif ceil(il/4)==blk && mod(il,4)==0
21 dk4=(double(ZZ(b+2,c+2)))-(double(ZZ(b+2,c+1)));
    cd(il,1)=il;
22 cd(il,2)=dk4;
23 cd(il,3)=ZZ(b+2,c+2);
24 cd(il,4)=ZZ(b+2,c+3);

25 end
26 end

```

Gambar 4.12 Kode Penyisipan Tahap Kedua

```

1  for il=115601:length(msga)
2  if (ceil((il-115600)/4))==blok1 && mod(ceil((il-
    115600)/4),4)==1
3  if abs(cd((il-115600),2))<=T
4  ZZ(b,c)=uint8(ZZ(b,c+1)+((2*(cd((il-
    115600),2)))+(msga(il))));
5  elseif abs(cd((il-115600),2))>T && (cd((il-115600),2))>=0
6  ZZ(b,c)=uint8(ZZ(b,c)+(T+1));
7  elseif abs(cd((il-115600),2))>T && (cd((il-115600),2))<0
8  ZZ(b,c)=uint8(ZZ(b,c)-(T+1));
9  end
10 ce(il,1)=il;

```

```

11 ce(il,2)=ZZ(b,c);
12
13 ...
14
15 elseif (ceil((il-115600)/4))==blok1 && mod(ceil((il-
    115600)/4),4)==0
16 if abs(cd((il-115600),2))<=T
17 ZZ(b+2,c+2)=uint8(ZZ(b+2,c+1)+(2*(cd((il-
    115600),2)))+(msga(il))));
18 elseif abs(cd((il-115600),2))>T && (cd((il-115600),2))>=0
19 ZZ(b+2,c+2)=uint8(ZZ(b+2,c+2)+(T+1));
20 elseif abs(cd((il-115600),2))>T && (cd((il-115600),2))<0
21 ZZ(b+2,c+2)=uint8(ZZ(b+2,c+2)-(T+1));
22 end

23 ce(il,1)=il;
24 ce(il,2)=ZZ(b+2,c+2);
25
26 end
27 end

```

4.1.4. Location Map

Berikut adalah kode yang digunakan untuk menentukan *Location Map* pada penyisipan tahap pertama.

Gambar 4.13 Kode untuk Menentukan *Location Map*

```

1 if -2<cz(ii,4,1) && cz(ii,4,1)<2
2 LM1=1;
3 elseif cz(ii,4,1)>=2 && mod(cz(ii,4,1),2)==1
4 LM1=2;
5 elseif cz(ii,4,1)>=2 && mod(cz(ii,4,1),2)==0
6 LM1=3;
7 elseif cz(ii,4,1)<=-2 && mod(cz(ii,4,1),2)==1
8 LM1=4;
9 elseif cz(ii,4,1)<=-2 && mod(cz(ii,4,1),2)==0
10 LM1=5;
11 end

```

4.2. Hasil Uji Coba dan Analisa

Hasil uji coba yang akan ditampilkan dibagi menjadi dua bagian, yaitu untuk citra *grayscale*. Masing-masing uji coba memiliki ... jenis citra sesuai dengan yang telah disebutkan pada bab sebelumnya.

4.2.1. Hasil Uji Coba Skenario 1

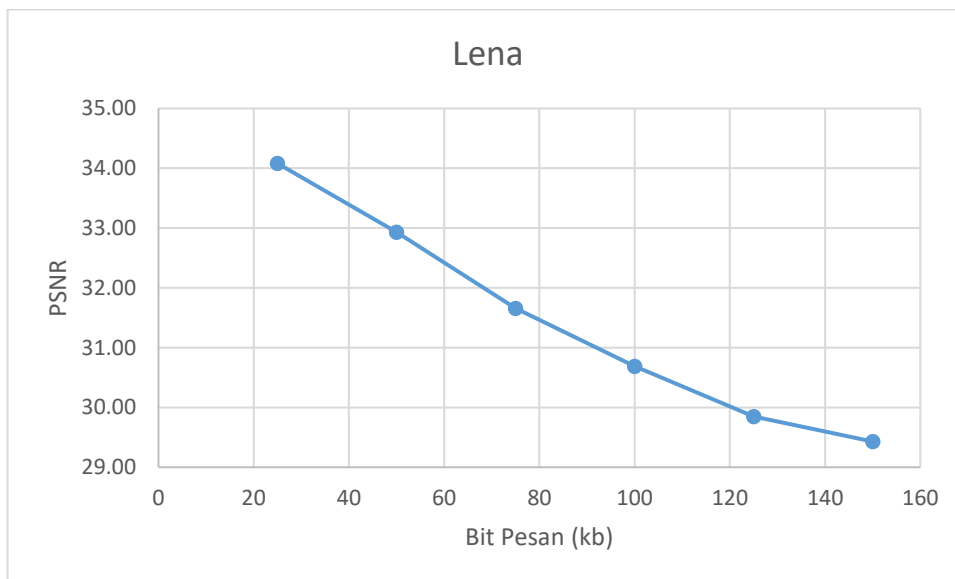
Pada percobaan ini setiap citra akan disisipi jumlah pesan dari 25 kilobit hingga 200 kilobit. Percobaan pertama akan dilakukan pada citra *grayscale* dan

mendapatkan parameter PSNR, SSIM, dan UQI dari citra Lena ditunjukkan pada Tabel 4.1. Sedangkan hasil citra yang telah disisipi pesan ditunjukkan pada Gambar 4.14.

Tabel 4.1 Hasil Penyisipan pada Citra Lena

Bit Pesan (kb)	PSNR	SSIM	UQI
25	34.08	0.9843	0.9999
50	32.93	0.9823	0.9999
75	31.66	0.9800	0.9999
100	30.69	0.9773	0.9998
125	29.85	0.9796	0.9999
150	29.43	0.9796	0.9999

Gambar 4.14 Grafik Perbandingan Kapasitas dan Kualitas Citra Lena

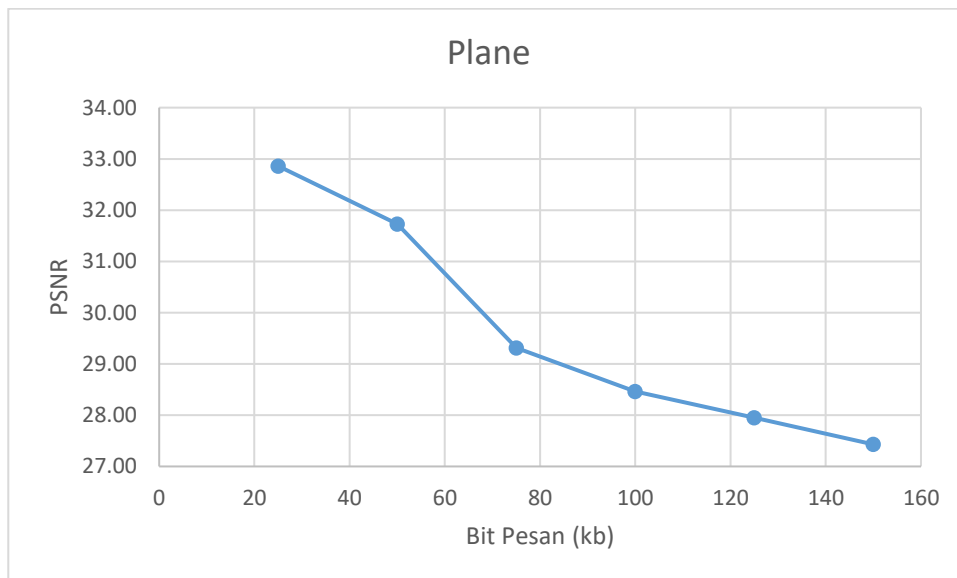


Pada percobaan ini setiap citra akan disisipi jumlah pesan dari 25 kilobit hingga kapasitas maksimal. Percobaan pertama akan dilakukan pada citra *grayscale* dan mendapatkan parameter PSNR, SSIM, dan UQI dari citra Plane ditunjukkan pada Tabel 4.2. Sedangkan grafik hasil citra yang telah disisipi pesan ditunjukkan pada Gambar 4.15.

Tabel 4.2 Hasil Penyisipan pada Citra Plane

Bit Pesan (kb)	PSNR	SSIM	UQI
25	32.86	0.9883	0.9999
50	31.73	0.9866	0.9999
75	29.31	0.9833	0.9999
100	28.46	0.9807	0.9998
125	27.95	0.9812	0.9999
150	27,43	0.9751	0.9981

Gambar 4.15 Grafik Perbandingan Kapasitas dan Kualitas Citra Plane



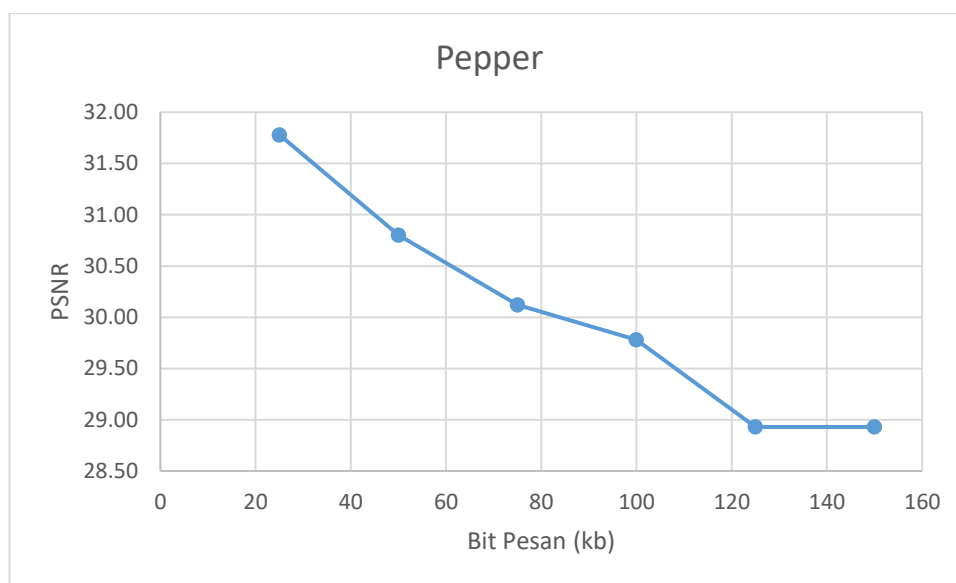
Pada percobaan ini setiap citra akan disisipi jumlah pesan dari 25 kilobit hingga 200 kilobit. Percobaan pertama akan dilakukan pada citra *grayscale* dan mendapatkan parameter PSNR, SSIM, dan UQI dar citra Pepper ditunjukkan pada Tabel 4.3. Sedangkan grafik hasil citra yang telah disisipi pesan ditunjukkan pada Gambar 4.16.

Tabel 4.3 Hasil Penyisipan pada Citra Pepper

Bit Pesan (kb)	PSNR	SSIM	UQI
25	31.78	0.9820	0.9966
50	30.80	0.9800	0.9980

75	30.12	0.9782	0.9985
100	29.78	0.9766	0.9995
125	28.93	0.9766	0.9978
150	28.93	0.9766	0.9996

Gambar 4.16 Grafik Perbandingan Kapasitas dan Kualitas Citra Pepper

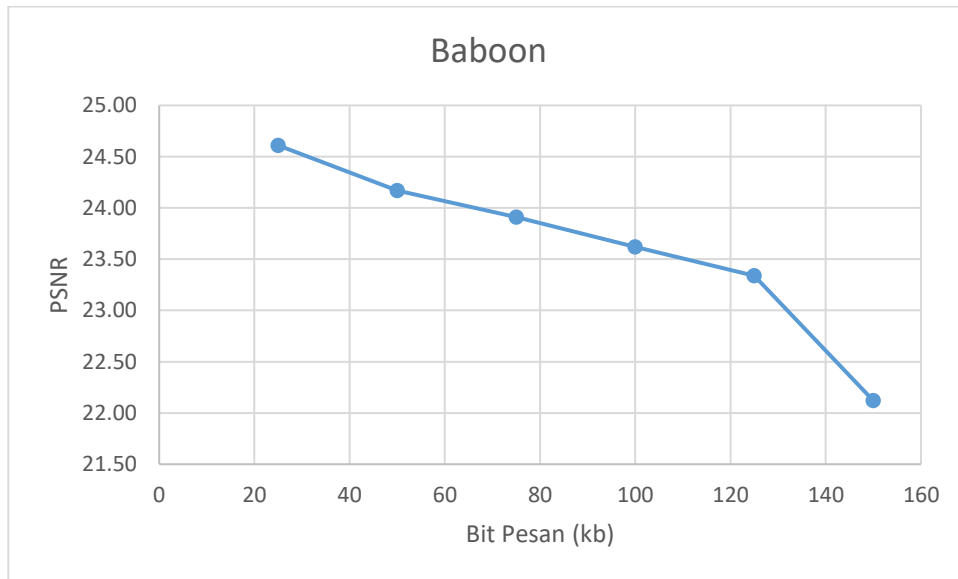


Pada percobaan ini setiap citra akan disisipi jumlah pesan dari 25 kilobit hingga 200 kilobit. Percobaan pertama akan dilakukan pada citra *grayscale* dan mendapatkan parameter PSNR, SSIM, dan UQI dari citra Baboon ditunjukkan pada Tabel 4.4. Sedangkan grafik hasil citra yang telah disisipi pesan ditunjukkan pada Gambar 4.17.

Tabel 4.4 Hasil Penyisipan pada Citra Baboon

Bit Pesan (kb)	PSNR	SSIM	UQI
25	24.61	0.9332	0.9993
50	24.17	0.9275	0.9991
75	23.91	0.9227	0.9990
100	23.62	0.9187	0.9988
125	23.34	0.9279	0.9990
150	22.12	0.9158	0.9815

Gambar 4.17 Grafik Perbandingan Kapasitas dan Kualitas Citra Baboon

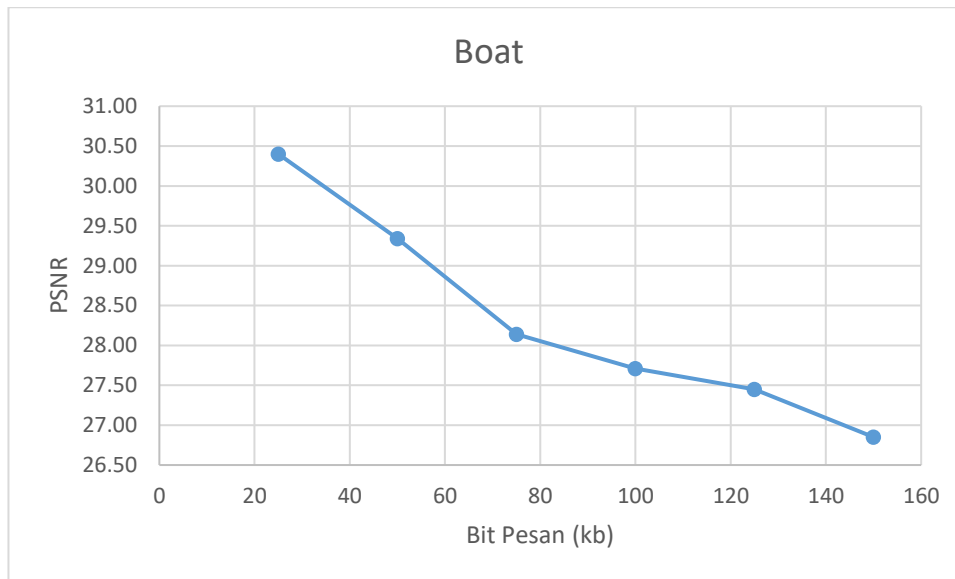


Pada percobaan ini setiap citra akan disisipi jumlah pesan dari 25 kilobit hingga 200 kilobit. Percobaan pertama akan dilakukan pada citra *grayscale* dan mendapatkan parameter PSNR, SSIM, dan UQI dari citra Baboon ditunjukkan pada Tabel 4.5. Grafik hasil citra yang telah disisipi pesan ditunjukkan pada Gambar 4.18.

Tabel 4.5 Hasil Penyisipan pada Citra Boat

Bit Pesan (kb)	PSNR	SSIM	UQI
25	30.40	0.9689	0.9983
50	29.34	0.9647	0.9983
75	28.14	0.9607	0.9983
100	27.71	0.9574	0.9988
125	27.45	0.9625	0.9984
150	26.85	0.9639	1.0202

Gambar 4.18 Penyisipan pada Citra Boat



4.2.2. Hasil Uji Coba Skenario 2

Pada percobaan ini dilakukan sebuah usaha steganalisis untuk setiap citra grayscale dengan ukuran pesan mulai dari 50 kb, 100 kb, hingga kapasitas maksimal masing-masing citra. Hasil percobaan skenario 2 akan ditunjukkan pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian Steganalysis pada Masing-masing Citra

Citra	50,000	100,000	Maksimal
Lena	6.626	6.784	6.876
Plane	6.905	6.432	6.578
Pepper	6.163	6.752	9.573
Boat	6.626	6.184	9.246
Baboon	6.163	6.652	9.893

4.2.3. Hasil Uji Coba Skenario 3

Pada percobaan ini dilakukan sebuah perubahan nilai pembobotan pada proses interpolasi. Daftar nilai pembobotan yang diubah dapat dilihat pada Tabel 4.7 yang menunjukkan pengubahan pembobotan pada blok piksel di pojok citra.

1	2	3	x
2			
3			
y			

Gambar 4.19 Posisi blok piksel di pojok citra

Posisi piksel yang mengalami perubahan pembobotan ditunjukkan pada Gambar 4.19. Sedangkan posisi piksel tersebut dalam blok ditunjukkan pada Gambar 4.20. Nilai pembobotan yang diubah dapat dilihat pada Persamaan 4.1 dan 4.2.

a	b	c	d	e	f
g	x	h	i	x	j
k	l	m	n	o	p
q	r	s	t	u	v
...	x	x	...
...

Gambar 4.20 Komponen blok piksel yang berada di pojok citra

$$IP_b = \left[\left(0.7 \times \left(\frac{a+c}{2} \right) \right) + (0.3 \times d) \right] \quad (4.1)$$

$$IP_g = \left[\left(0.7 \times \left(\frac{a+k}{2} \right) \right) + (0.3 \times q) \right] \quad (4.2)$$

Tabel 4.7 Uji Coba Nilai Pembobotan pada Sisi Pojok

No.	Posisi Piksel	Nilai Orisinal	Percobaan 1	Percobaan 2
1.	Sisi pojok	0.7	0.8	0.9
		0.3	0.2	0.1
2.	Sisi luar	0.5	0.5	0.5
		0.25	0.25	0.25
		0.25	0.25	0.25
3.	Sisi tepi dalam	0.5	0.5	0.5
		0.3	0.3	0.3

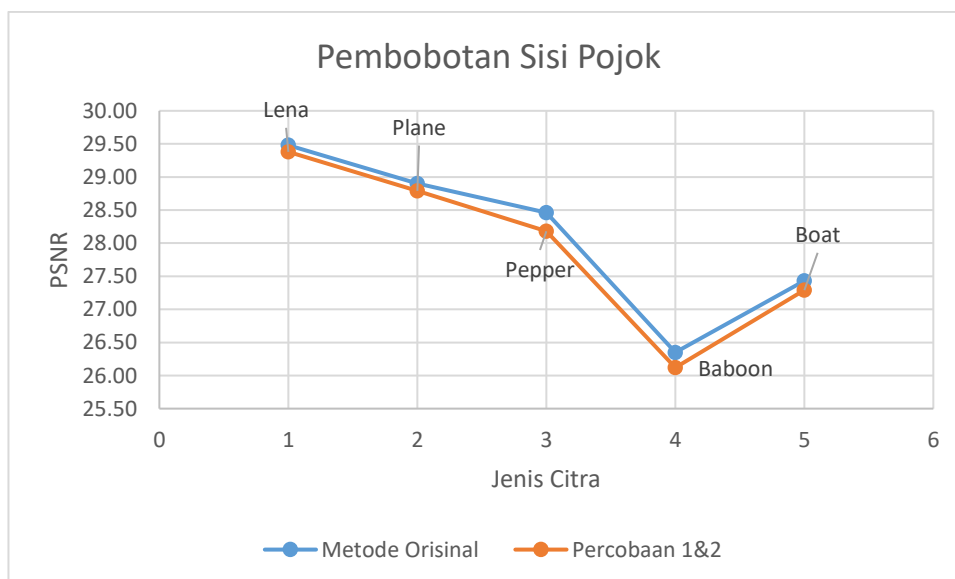
		0.2	0.2	0.2
4.	Sisi tengah	0.5	0.5	0.5
		0.3	0.3	0.3
		0.2	0.2	0.2

Hasil dari pengubahan pembobotan dapat dilihat pada Tabel 4.8, untuk lebih jelasnya dapat memeriksa Gambar 4.21.

Tabel 4.8 Hasil PSNR dari Pengubahan Pembobotan Sisi Pojok

No	Citra	Orisinal	Percobaan 1	Percobaan 2
1.	Lena	29.48	29.38	29.38
2.	Plane	28.90	28.79	28.79
3.	Pepper	28.46	28.18	28.18
4.	Baboon	26.35	26.12	26.12
5.	Boat	27.43	27.29	27.29

Gambar 4.21 Grafik Pengubahan Pembobotan Sisi Pojok



1	2	3	x
2			
3			
y			

Gambar 4.22 Posisi blok piksel di sisi citra

Posisi piksel yang mengalami perubahan pembobotan ditunjukkan pada Gambar 4.22. Sedangkan posisi piksel tersebut dalam blok ditunjukkan pada Gambar 4.23. Nilai pembobotan yang diubah dapat dilihat pada Persamaan 4.3.

a	b	c	d	e	f	q	r	s
g	x	h	i	x	j	...	x	...
k	l	m	n	o	p
...	...	t	u	v	w
...	x	x	x	...
...

Gambar 4.23 Komponen blok piksel yang berada di sisi citra

$$IP_e = \left\lfloor \left(0.5 \times \left(\frac{d+f}{2} \right) \right) + (0.25 \times c) + (0.25 \times q) \right\rfloor \quad (4.3)$$

Tabel 4.9 Uji Coba Nilai Pembobotan pada Sisi Luar

No.	Posisi Piksel	Nilai Orisinal	Percobaan 1	Percobaan 2	Percobaan 3
1.	Sisi pojok	0.7	0.7	0.7	0.7
		0.3	0.3	0.3	0.3
2.	Sisi luar	0.5	0.6	0.7	0.8
		0.25	0.2	0.15	0.1
		0.25	0.2	0.15	0.1
3.	Sisi tepi dalam	0.5	0.5	0.5	0.5
		0.3	0.3	0.3	0.3
		0.2	0.2	0.2	0.2
4.	Sisi tengah	0.5	0.5	0.5	0.5

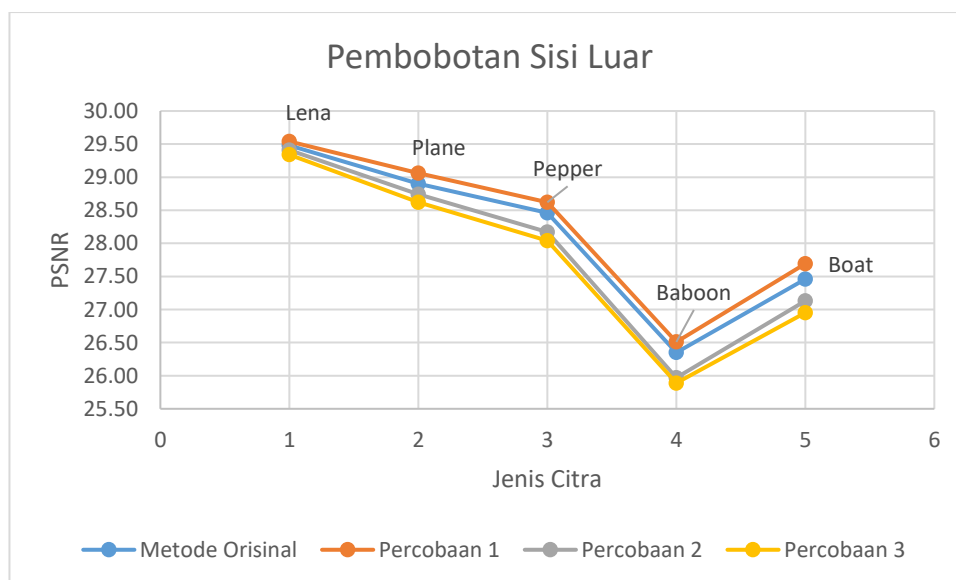
		0.3	0.3	0.3	0.3
		0.2	0.2	0.2	0.2

Tabel 4.10 Hasil PSNR dari Pengubahan Pembobotan Sisi Luar

No	Citra	Orisinal	Percobaan 1	Percobaan 2	Percobaan 3
1.	Lena	29.48	29.54	29.41	29.34
2.	Plane	28.90	29.06	28.74	28.62
3.	Pepper	28.46	28.62	28.17	28.04
4.	Baboon	26.35	26.51	25.97	25.89
5.	Boat	27.46	27.69	27.13	26.95

Pada Tabel 4.9 adalah pengubahan nilai pembobotan pada piksel-piksel yang berada di sisi luar citra. Hasilnya ditunjukkan pada Tabel 4.10 dan Gambar 4.24.

Gambar 4.24 Grafik Pengubahan Pembobotan Sisi Luar



Sedangkan pada Tabel 4.11 merupakan uji coba pengubahan nilai pembobotan untuk piksel-piksel yang berada pada sisi tepi dalam. Posisi piksel yang mengalami pengubahan pembobotan ditunjukkan pada Gambar 4.22.

Sedangkan posisi piksel tersebut dalam blok ditunjukkan pada Gambar 4.23. Nilai pembobotan yang diubah dapat dilihat pada Persamaan 4.4. Hasil pengujian dapat dilihat pada Tabel 4.12 dan Gambar 4.25 mengenai PSNR masing-masing citra.

$$IP_i = \left[\left(0.5 \times \left(\frac{d+n}{2} \right) \right) + \left(0.3 \times \left(\frac{c+m}{2} \right) \right) + \left(0.2 \times \left(\frac{t+u}{2} \right) \right) \right] \quad (4.4)$$

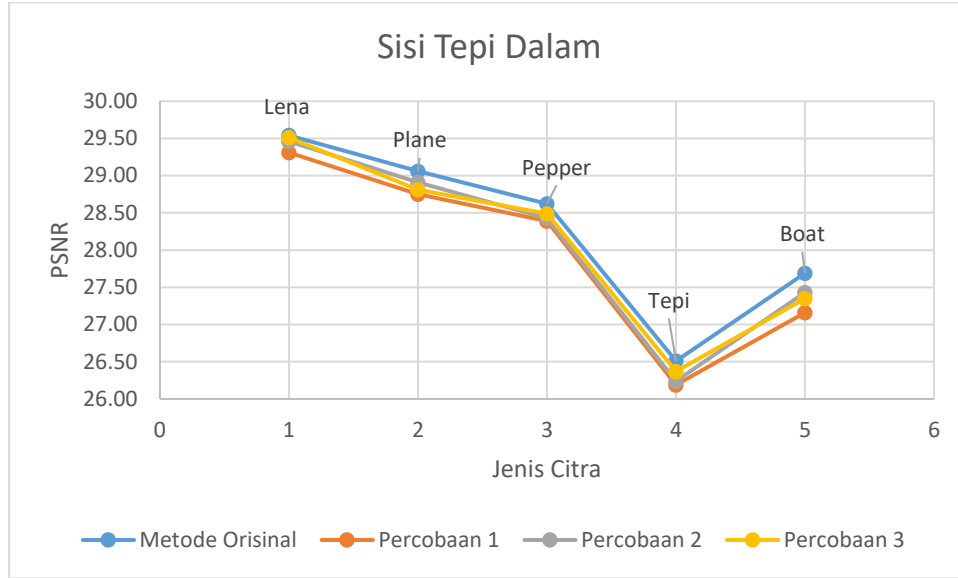
Tabel 4.11 Uji Coba Nilai Pembobotan pada Sisi Tepi Dalam

No.	Posisi Piksel	Nilai Orisinal	Percobaan 1	Percobaan 2	Percobaan 3
1.	Sisi pojok	0.7	0.7	0.7	0.7
		0.3	0.3	0.3	0.3
2.	Sisi luar	0.6	0.6	0.6	0.6
		0.2	0.2	0.2	0.2
		0.2	0.2	0.2	0.2
3.	Sisi tepi dalam	0.5	0.6	0.7	0.8
		0.3	0.25	0.2	0.15
		0.2	0.15	0.1	0.05
4.	Sisi tengah	0.5	0.5	0.5	0.5
		0.3	0.3	0.3	0.3
		0.2	0.2	0.2	0.2

Tabel 4.12 Hasil PSNR dari Pengubahan Pembobotan Sisi Tepi Dalam

No	Citra	Orisinal	Percobaan 1	Percobaan 2	Percobaan 3
1.	Lena	29.54	29.31	29.46	29.51
2.	Plane	29.06	28.75	28.91	28.81
3.	Pepper	28.62	28.39	28.43	28.49
4.	Baboon	26.51	26.19	26.24	26.37
5.	Boat	27.69	27.16	27.43	27.35

Gambar 4.25 Grafik Pengubahan Pembobotan Sisi Sisi Tepi Dalam



1	2	3	x
2			
3			
y			

Gambar 4.26 Posisi blok piksel di tengah-tengah

Posisi piksel yang mengalami perubahan pembobotan ditunjukkan pada Gambar 4.26. Sedangkan posisi piksel tersebut dalam blok ditunjukkan pada Gambar 4.27. Nilai pembobotan yang diubah dapat dilihat pada Persamaan 4.5.

a	b	c	d	e	f
g	x	h	...	x	x	...
k	l	m	n	o	p	t	u	v
...	...	i	q	r	s	j
...	x	x	x	...
...

Gambar 4.27 Komponen blok piksel yang berada di tengah citra

$$IP_o = \left\lfloor \left(0.5 \times \left(\frac{n+p}{2} \right) \right) + \left(0.3 \times \left(\frac{q+s}{2} \right) \right) + \left(0.2 \times \left(\frac{m+i+t+j}{4} \right) \right) \right\rfloor \quad (4.5)$$

Tabel 4.13 Uji Coba Nilai Pembobotan pada Sisi Tengah

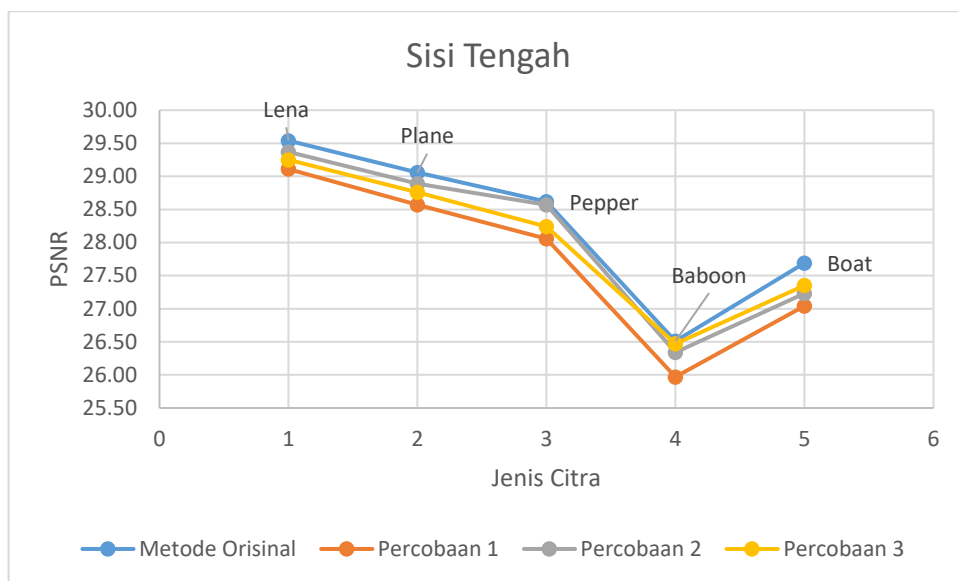
No.	Posisi Piksel	Nilai Orisinal	Percobaan 1	Percobaan 2	Percobaan 3
1.	Sisi pojok	0.7	0.7	0.7	0.7
		0.3	0.3	0.3	0.3
2.	Sisi luar	0.6	0.6	0.6	0.6
		0.2	0.2	0.2	0.2
		0.2	0.2	0.2	0.2
3.	Sisi tepi dalam	0.5	0.5	0.5	0.5
		0.3	0.3	0.3	0.3
		0.2	0.2	0.2	0.2
4.	Sisi tengah	0.5	0.6	0.7	0.8
		0.3	0.25	0.2	0.15
		0.2	0.15	0.1	0.05

Pada Tabel 4.13 adalah pengubahan nilai pembobotan pada piksel-piksel yang berada di sisi tengah citra. Hasilnya ditunjukkan pada Tabel 4.14 dan Gambar 4.28.

Tabel 4.14 Hasil PSNR dari Pengubahan Pembobotan Sisi Tengah

No	Citra	Orisinal	Percobaan 1	Percobaan 2	Percobaan 3
1.	Lena	29.54	29.11	29.37	29.25
2.	Plane	29.06	28.57	28.89	28.76
3.	Pepper	28.62	28.06	28.57	28.24
4.	Baboon	26.51	25.97	26.34	26.47
5.	Boat	27.69	27.04	27.23	27.35

Gambar 4.28 Grafik Perubahan Pembobotan Sisi Tengah



4.2.4. Hasil Uji Coba Skenario 4

Pada percobaan ini dilakukan penggantian nilai *thresholding* untuk mengetahui jumlah kapasitas maksimal setiap iterasi dalam citra. Hasil pengujian ditunjukkan pada Tabel 4.15.

Tabel 4.15 Hasil Uji Coba Penggantian Nilai *Thresholding*

Citra	T=4			T=6		
	Kapasitas	PSNR	SSIM	Kapasitas	PSNR	SSIM
Lena	193,649	29.48	0.9766	212,328	29.20	0.9774
Plane	181,542	28.90	0.9654	195,102	28.57	0.9647
Pepper	166,563	28.46	0.9582	187,873	27.90	0.9526
Baboon	193,124	26.35	0.9431	201,983	24.78	0.9125
Boat	176,456	27.46	0.9481	189,674	26.12	0.9348
T=10				T=14		
Lena	232,084	28.89	0.9762	241,817	28.46	0.9762
Plane	205,489	27.72	0.9796	219,360	26.95	0.9796
Pepper	214,328	27.53	0.9755	221,573	26.87	0.9854
Baboon	206,890	22.52	0.9158	209,161	21.90	0.9158
Boat	200,156	25.75	0.9246	228,735	25.03	0.9216

4.3. Hasil Uji Coba *Neighbor Mean Interpolation* (NMI)

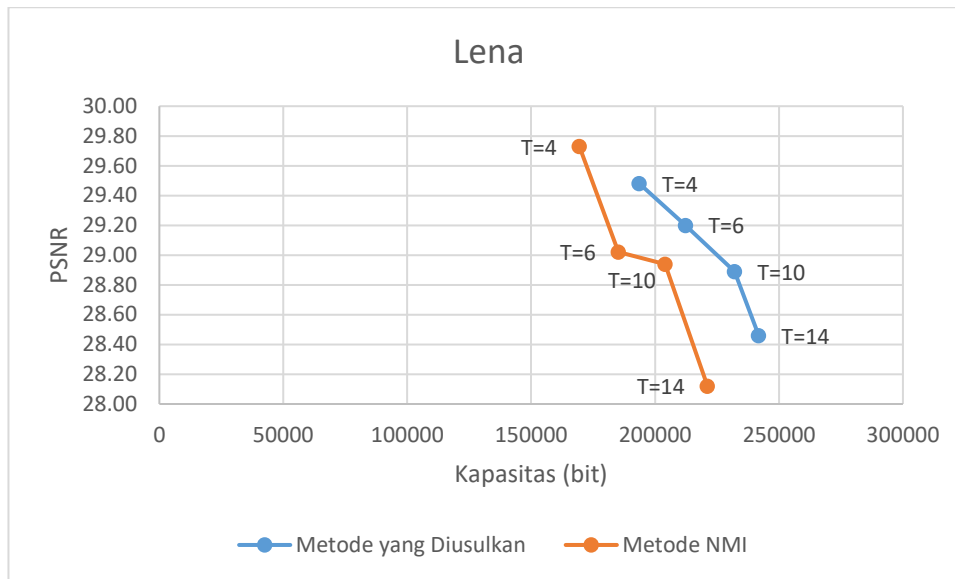
Berikut adalah hasil uji coba Neighbor Mean Interpolation (NMI) yang dikombinasikan dengan proses penyisipan dua tahap yang telah dilakukan pada uji coba sebelumnya. (Jung, 2009)

Tabel 4.16 Hasil Uji Coba Penyisipan Pesan dengan NMI

Citra	T=4			T=6		
	Kapasitas	PSNR	SSIM	Kapasitas	PSNR	SSIM
Lena	169,453	29.73	0.9845	185,202	29.02	0.9766
Plane	179,515	27.85	0.9412	185,205	27.27	0.8569
Pepper	168,430	28.14	0.9547	185,091	27.37	0.8100
Baboon	183,808	27.34	0.9345	193,105	26.69	0.9125
Boat	150,676	27.86	0.9354	174,372	26.69	0.9298
T=10				T=14		
Lena	204,084	28.94	0.9645	221,147	28.12	0.9641
Plane	196,581	26.96	0.9354	207,159	26.04	0.9246
Pepper	205,941	27.02	0.8398	212,363	26.53	0.8247
Baboon	203,849	25.94	0.9158	205,918	24.51	0.9158
Boat	192,167	25.54	0.9164	205,143	24.91	0.9124

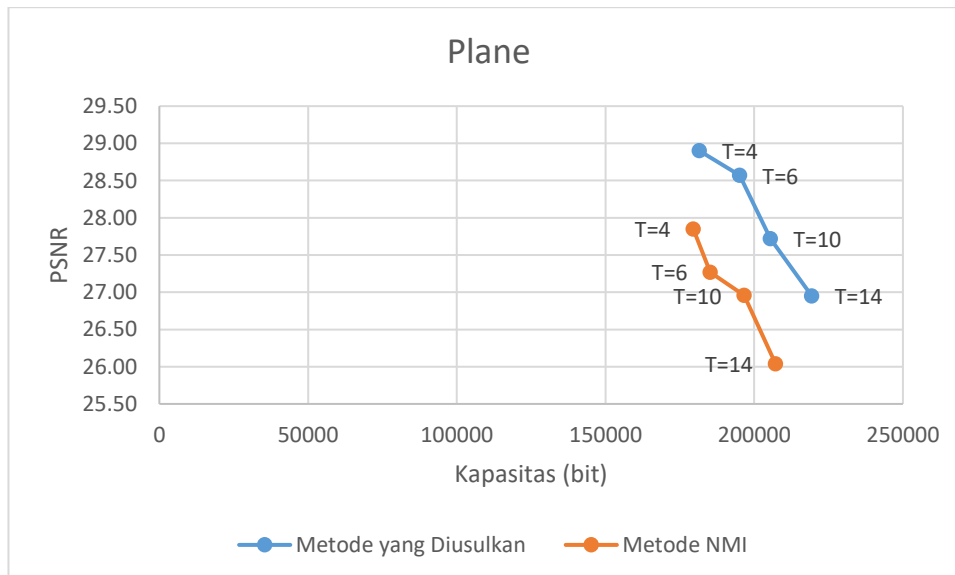
Berikut adalah grafik perbandingan kualitas dan kuantitas antara dua metode interpolasi yang digunakan. Grafik-grafik tersebut berdasarkan hasil dari Tabel 4.15 dan 4.16. Perbandingan antara kualitas dan kuantitas dari citra Lena berdasarkan nilai *thresholding* ditunjukkan pada Gambar 4.29.

Gambar 4.29 Grafik Perbandingan dengan Metode NMI Citra Lena

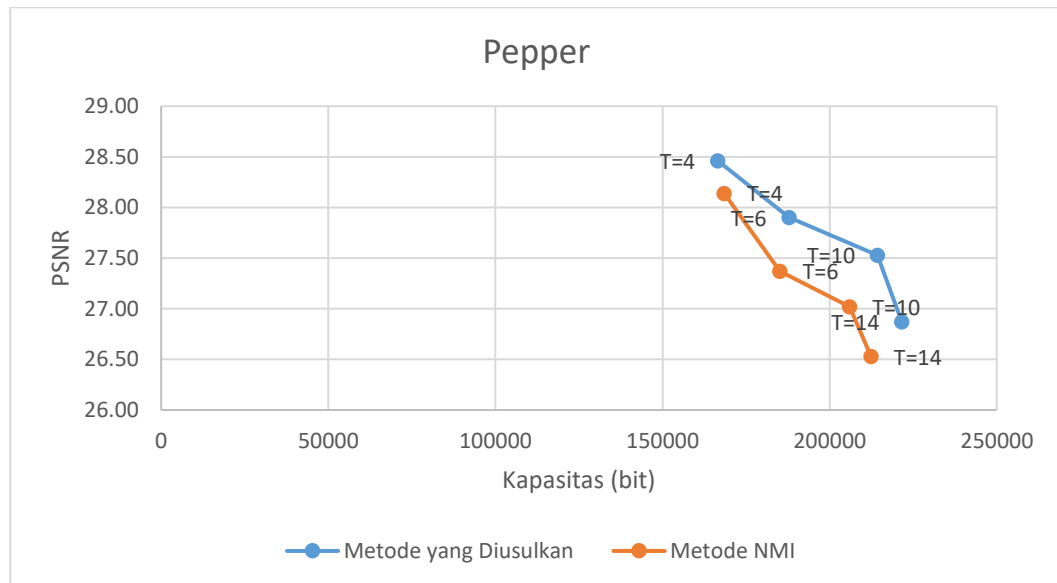


Sedangkan perbandingan antara kualitas dan kuantitas metode yang diusulkan dengan metode kombinasi dari citra Plane ditunjukkan pada Gambar 4.30. Gambar 4.31 menunjukkan perbandingan tersebut khusus untuk citra Pepper.

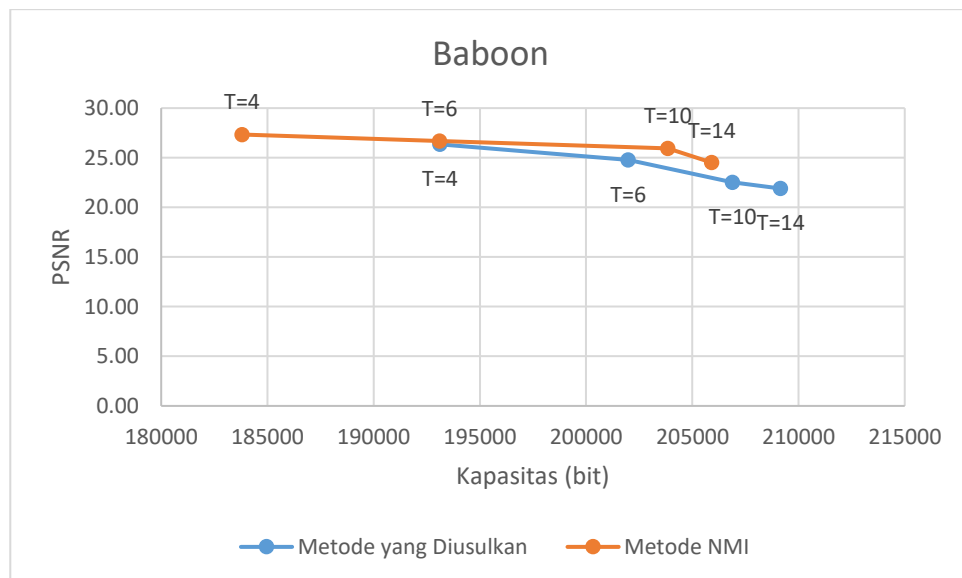
Gambar 4.30 Grafik Perbandingan dengan Metode NMI Citra Plane



Gambar 4.31 Grafik Perbandingan dengan Metode NMI Citra Pepper

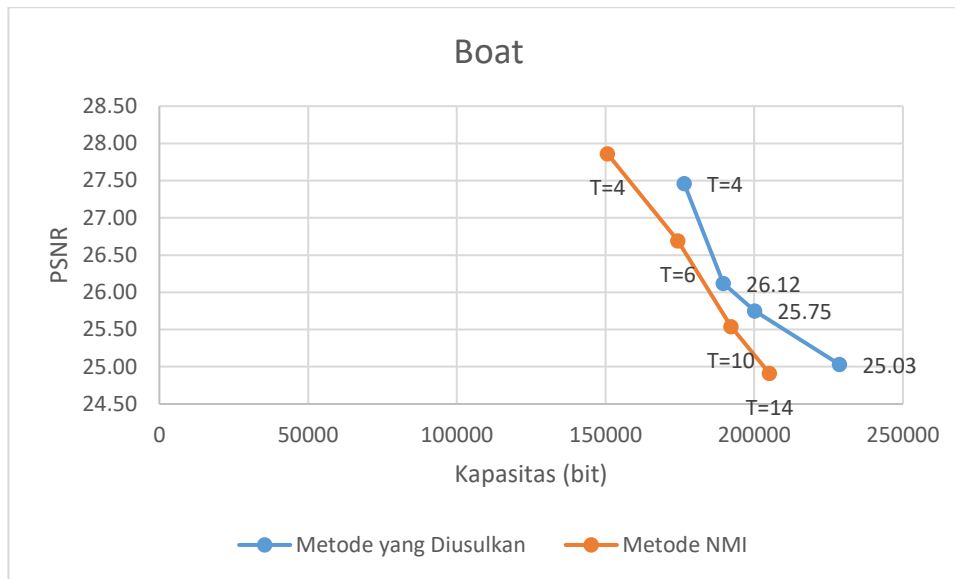


Gambar 4.32 Grafik Perbandingan dengan Metode NMI Citra Baboon



Perbandingan antara kualitas dan kuantitas dari citra Baboon berdasarkan nilai *thresholding* ditunjukkan pada Gambar 4.32. Sedangkan perbandingan antara kualitas dan kuantitas dari citra Boat berdasarkan nilai *thresholding* ditunjukkan pada Gambar 4.33.

Gambar 4.33 Grafik Perbandingan dengan Metode NMI Citra Boat



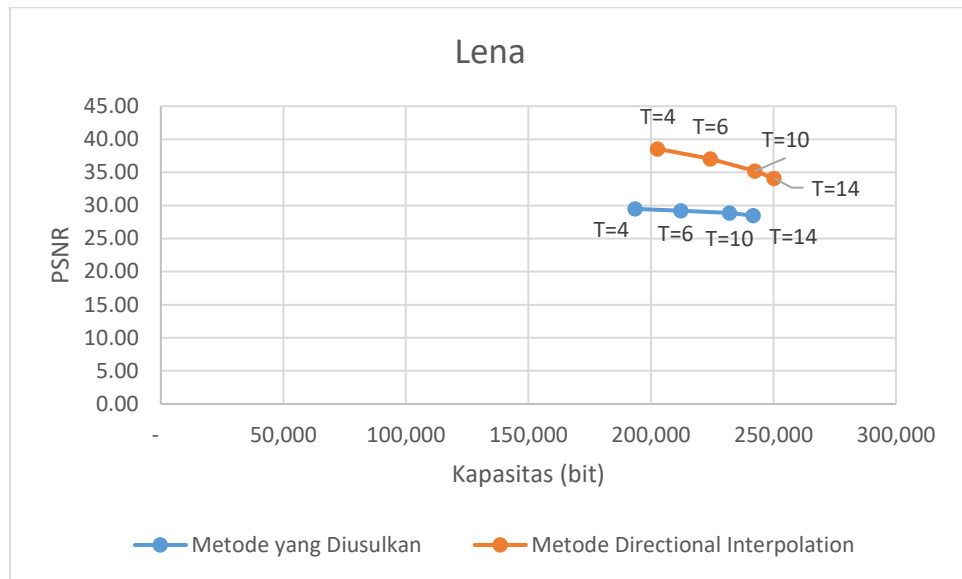
4.4. Hasil Uji Coba *Directional Interpolation and Difference Expansion*

Sedangkan berikut adalah hasil dari proses interpolasi *Directional Interpolation* dan *Difference Expansion*. (Govind, 2015)

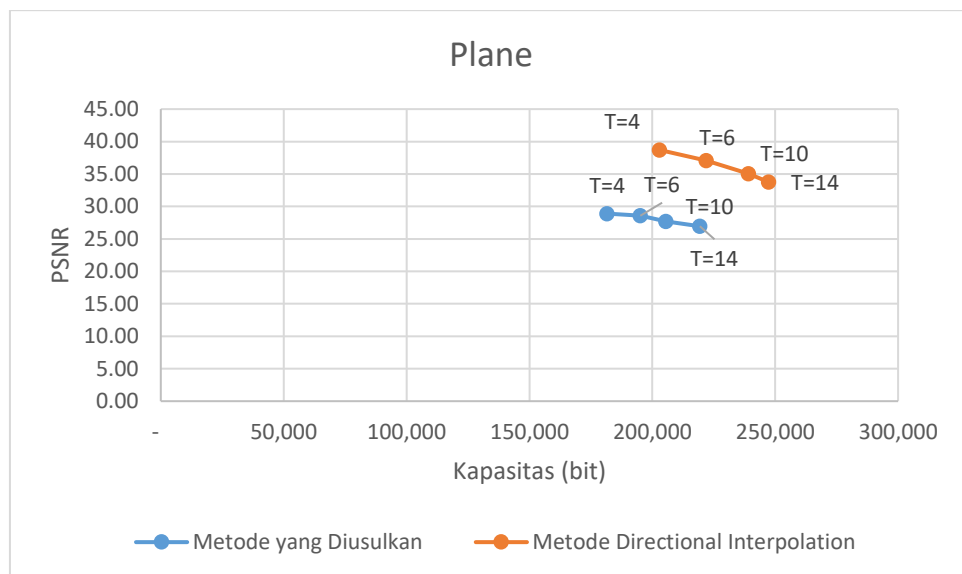
Tabel 4.17 Hasil Uji Coba *Directional Interpolation* dan DE

Citra	T=4			T=6		
	Kapasitas	PSNR	SSIM	Kapasitas	PSNR	SSIM
Lena	202.780	38.56	0.9923	224.306	37.02	0.9909
Plane	202.878	38.71	0.9922	221.959	37.06	0.9931
Boat	169.902	37.32	0.9939	194.802	35.40	0.9924
T=10			T=14			
Lena	242.437	35.20	0.9893	250.343	34.09	0.9885
Plane	239.180	35.03	0.9916	247.297	33.76	0.9906
Boat	220.869	33.16	0.9899	233.918	31.60	0.9883

Gambar 4.34 Grafik Perbandingan Metode DI Citra Lena

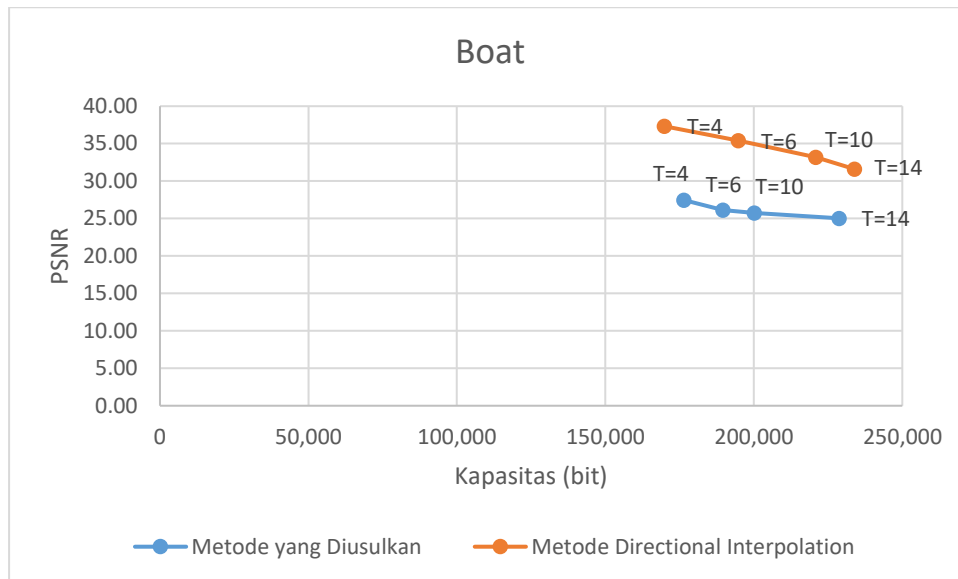


Gambar 4.35 Grafik Perbandingan Metode DI Citra Plane



Grafik perbandingan antara dua metode, yaitu metode yang diusulkan dengan metode Directional Interpolation. Pada Gambar 4.33 adalah perbandingan kedua metode berdasarkan citra Lena, sedangkan Gambar 4.35 untuk citra Plane, dan Gambar 4.36 untuk citra Boat.

Gambar 4.36 Grafik Perbandingan Metode DI Citra Boat



4.5. Analisa Hasil Percobaan

Proses penyisipan yang dilakukan pada citra Lena hingga Boat ditunjukkan pada Tabel 4.1 hingga Tabel 4.5. Proses penyisipan dilakukan setiap 25 kilobit hingga 150 kilobit. Penurunan nilai PSNR terjadi secara bertahap pada masing-masing citra dengan nilai yang stabil berbanding terbalik dengan kapasitas penyimpanan pesan.

Nilai *threshold* memiliki pengaruh pada selisih antara piksel hasil interpolasi dengan piksel citra orisinal. Hal ini disebabkan oleh jumlah piksel yang bisa disisipi pesan pada proses penyisipan tahap dua bergantung pada nilai selisih yang tidak lebih dari nilai *threshold* yang telah ditentukan. Sehingga semakin tinggi nilai *threshold*, maka semakin banyak pula kapasitas pesan yang dapat disisipkan.

Semakin besar nilai KL-Distance yang dimiliki oleh masing-masing citra stego yang ditunjukkan pada Tabel 4.6, maka semakin besar pula kemungkinan adanya pesan rahasia. Untuk citra Lena dan Plane memiliki nilai lebih kecil karena jangkauan warna yang dimiliki berada di tengah-tengah. Namun untuk citra Baboon dan Boat yang memiliki warna lebih gelap, maka semakin sedikit juga jumlah piksel yang dapat disimpan.

Uji coba pengubahan nilai pembobotan dilakukan secara berulang pada masing-masing area hingga dicari mana nilai yang paling maksimal. Percobaan

pertama dilakukan pada Sisi Pojok. Pada Gambar 4.19 dapat dilihat tidak terdapat selisih yang signifikan pada kualitas citra, namun nilai yang paling maksimal terdapat pada nilai pembobotan yang diajukan. Hal ini disebabkan posisi nilai piksel di pojok jumlahnya sangat sedikit sehingga tidak terlalu mempengaruhi perubahan nilai. Maka nilai bobot yang dipilih adalah nilai orisinal.

Sedangkan uji coba pada area tepi luar ditunjukkan pada Gambar 4.20 menunjukkan bahwa nilai pembobotan pada percobaan dua merupakan nilai yang paling maksimal. Sehingga nilai tersebut yang dipilih sebagai nilai pembobotan maksimal.

Percobaan pencarian nilai pembobotan pada sisi tepi dalam dapat ditunjukkan pada Gambar 4.21, nilai pembobotan dengan kualitas paling maksimal adalah nilai pembobotan yang diajukan.

Uji coba pencarian nilai pembobotan pada sisi tengah ditunjukkan pada Gambar 4.22. Terdapat selisih nilai kualitas yang sangat sedikit antara nilai pembobotan yang diajukan dengan nilai pembobotan dari uji coba lainnya. Sehingga nilai pembobotan maksimal yang dipilih adalah nilai yang diajukan.

Dari Tabel 4.15 dapat disimpulkan dari iterasi perubahan nilai *thresholding* dari 10 ke 14, terdapat penambahan bit data kurang lebih sebanyak 10,000 bit data. Sedangkan untuk iterasi perubahan nilai *thresholding* dari 4 ke 6 terdapat penambahan bit data kurang lebih sebanyak 15-20 kilobit data. Dapat dilihat pada Gambar 4.23 hingga 4.27 perbandingan nilai PSNR antar metode interpolasi yang berbeda untuk masing-masing citra.

Sedangkan pada Tabel 4.16 merupakan hasil uji coba metode interpolasi NMI yang dikombinasikan dengan metode penyisipan pesan dua kali seperti metode yang diajukan. Dapat dibandingkan dengan hasil perubahan nilai *thresholding* pada Tabel 4.15, terdapat selisih kapasitas yang cukup banyak antara 15 hingga 20 kilobit untuk setiap iterasi.

Perbandingan hasil juga dilakukan dengan percobaan Interpolasi Direksional (Govind, 2015) yang ditunjukkan pada Tabel 4.17, kapasitas dan PSNR dari metode yang diusulkan memiliki selisih 10 hingga 20 kilobit lebih rendah. Hal ini disebabkan oleh penghitungan pembobotan piksel-piksel pada DI menggunakan

prediksi error yang lebih bervariasi sehingga menghasilkan nilai prediksi yang lebih baik daripada menggunakan rumus rata-rata biasa.

Sedangkan pada metode interpolasi yang diajukan hanya menggunakan rumus rata-rata yang digunakan pada metode interpolasi dasar dengan pengembangan jenis pembobotan berdasarkan posisi piksel yang diinterpolasi terhadap piksel-piksel yang bertetangga.

BAB 5

KESIMPULAN DAN SARAN

Setelah penelitian yang telah dilakukan, dapat diberikan kesimpulan dan saran yang dapat digunakan untuk mengembangkan penelitian ini. Saran-saran yang diberikan digunakan untuk pengembangan metode ini dalam penelitian selanjutnya.

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Penurunan nilai PSNR terjadi secara bertahap pada masing-masing citra dengan nilai yang stabil berbanding terbalik dengan kapasitas penyimpanan pesan.
2. Citra Lena dan Plane memiliki nilai *KL-Distance* lebih kecil karena jangkauan warna yang dimiliki berada di tengah-tengah. Namun untuk citra Baboon dan Boat yang memiliki warna lebih gelap, maka semakin sedikit juga jumlah piksel yang dapat disimpan.
3. Iterasi perubahan nilai *thresholding* dari 10 ke 14, terdapat penambahan bit data kurang lebih sebanyak 10,000 bit data. Sedangkan untuk iterasi perubahan nilai *thresholding* dari 4 ke 6 terdapat penambahan bit data kurang lebih sebanyak 15-20 kilobit data.
4. Hasil Interpolasi NMI dengan metode penyisipan pesan dua kali seperti metode yang diajukan menghasilkan perubahan nilai selisih kapasitas yang cukup banyak antara 15 hingga 20 kilobit untuk setiap iterasi.
5. Kapasitas dan PSNR dari metode yang diusulkan memiliki selisih 10 hingga 20 kilobit lebih rendah dibandingkan metode Interpolasi Direksional. Hal ini disebabkan oleh penghitungan pembobotan piksel-piksel pada DI menggunakan prediksi *error* yang lebih bervariasi sehingga menghasilkan nilai prediksi yang lebih baik daripada menggunakan rumus rata-rata biasa.

6. Metode yang diajukan dapat menambah kompleksitas proses penyisipan pesan rahasia yang telah ada saat ini.
7. Metode yang diajukan dapat menambah kapasitas pesan rahasia dibandingkan beberapa metode penyisipan pesan yang telah ada.
8. Metode yang diajukan tidak dapat meningkatkan kualitas citra stego sebagai konsekuensi dari penggunaan piksel secara maksimal.
9. Pengembangan teknik interpolasi bertujuan untuk mencari nilai selisih yang paling minimal untuk menambah jumlah piksel yang dapat disisipkan pada citra cover.

5.2. Saran

Saran yang dapat diberikan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Penggunaan piksel secara maksimal dapat dilakukan dengan eksperimen pada pembatasan jumlah piksel pada sebuah blok.
2. Penambahan kapasitas dapat dilakukan dengan beberapa kali proses penyisipan pada piksel citra stego.
3. Percobaan untuk memaksimalkan kapasitas dapat dilakukan dengan mengembangkan proses penghitungan pembobotan pada setiap piksel dalam proses interpolasi yang diusulkan berdasarkan posisi piksel-piksel yang bertetangga.

DAFTAR PUSTAKA

- Ahmad, T. & H. M., 2014. Peningkatan Performa Metode Steganografi Berbasis Difference Expansion Menggunakan Reduksi Selisih. *JUTI : Jurnal Ilmiah Teknologi Informasi*, pp. 9-17.
- Alattar, A. M., 2004. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Transactions on Image Processing*, pp. 1147-1156.
- Almohammad, A. G. G., 2010. Stego image quality and the reliability of PSNR. *2010 2nd International Conference on Image Processing Theory, Tools and Applications, IPTA 2010*, pp. 215-220.
- Barton, J. M., 1997. "Method and apparatus for embedding authentication information within digital data.". U.S., Patent No. 5,646,997.
- Carolina, U. o. S., 2015. *SIPi Image Database*. [Online]
Available at: <http://sipi.usc.edu/database/database.php?volume=misc>
[Accessed 5 February 2016].
- Celik, M. U. G. S. A. M. T. d. E. S., 2002. *Reversible data hiding*. s.l., IEEE, pp. II-157.
- De Vleeschouwer, C. d. M. B., 2003. Circular interpretation of bijective transformations in lossless watermarking for media asset management.. *IEEE Transactions on Multimedia*, 5(1), pp. 97-105.
- Diljith, M. T. & Jeffrey, J. R., 2007. Expansion Embedding Techniques for Reversible Watermarking. *IEEE Transactions on Image Processing*, 16(3), pp. 721-730.
- Fridrich, J. M. G. d. R. D., 2002. Lossless data embedding: new paradigm in digital watermarking.. *EURASIP Journal on Applied Signal Processing*, pp. 185-196.
- Govind, P. S. & W. M., 2015. A New Reversible Data Hiding Scheme with Improved Capacity Based on Directional Interpolation and Difference Expansion. *Procedia Computer Science*, pp. 491-498.

- Honsinger, C. W. P. W. J. M. R. d. J. C. S., 2001. *"Lossless recovery of an original image containing embedded data."*. U.S., Patent No. 6,278,791.
- Husrev T. Sencar, M. R. & A. N. A., 2004. *Data hiding fundamentals and applications: content security in digital multimedia*. s.l.:Academic Press.
- Jung, K.-H. K.-Y. Y., 2009. Data hiding method using image interpolation. *Computer Standards & Interfaces*, pp. 465-470.
- Kalker, T. O. N. d. W. F. M., 2003. *"Capacity bounds and constructions for reversible data-hiding."*. s.l., International Society for Optics and Photonics, pp. 604-611.
- Library, E., 2015. *Partners Infectious Disease*. [Online]
Available at: <http://www.idimages.org/images/browse/ImageTechnique/>
[Accessed 01 Februari 2016].
- Lou, D. C. H. M. & L. J., 2009. Multiple layer data hiding scheme for medical images. *Computer Standards & Interfaces*, Februari.pp. 329-335.
- Macq, B., 2000. Lossless multiresolution transform for image authenticating watermarking. *Proceeding EUSPICO*, pp. 533-536.
- Tian, J., 2003. Reversible Data Embedding Using a Difference Expansion. *IEEE Transactions on Circuits and Systems*, pp. 890-896.
- Wang, Z. A. C. B. H. R. S. E. P. S., 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, pp. 600-612.
- Zhou Wang, A. C. B., 2002. A Universal Image Quality Index. *IEEE Signal Processing Letters*, March, 9(3), pp. 81-84.

BIODATA PENULIS



Lidya Amalia Rahmania atau biasa dipanggil Lidya dilahirkan di Malang pada tanggal 9 Oktober 1990. Puteri dari Mulyono, S.AB., dan Nuzula Mafruchah, S.AB., M.M. Penulis adalah anak pertama dari tiga bersaudara. Adik pertama penulis, Fidya Taufiqi Abdillah, S.IP., telah menempuh pendidikan Ilmu Hubungan Internasional di Universitas Brawijaya. Adik kedua penulis, Sermatutar Arm. M. Hafizh Al Maliki, S.T.Han., telah menempuh pendidikan militer di Akademi Militer Magelang.

Penulis menempuh pendidikan di SD Negeri Kauman I Malang (1996-2002), kemudian melanjutkan ke SMP Negeri 3 Malang (2002-2005), dan SMA Negeri 1 Malang (2005-2008). Mendapatkan gelar Sarjana Komputer setelah menempuh pendidikan di Teknik Informatika Universitas Brawijaya Malang (2008-2013). Menyelesaikan studi Magister di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya (2014-2017). Bidang studi yang diambil adalah Komputasi Berbasis Jaringan.

Selama menempuh pendidikan sarjana penulis aktif mendirikan dan menjadi anggota Himpunan Mahasiswa Teknik Informatika (HMIF UB). Selain itu penulis merupakan asisten dosen dan asisten laboratorium Jaringan Komputer sehingga menumbuhkan minat berbagi ilmu penulis dan menjadi motivasi untuk menempuh pendidikan pasca sarjana.

[Halaman ini sengaja dikosongkan]